A high-order Discontinuous Galerkin solver for incompressible and low-Mach number flows

Vom Fachbereich Maschinenbau
an der Technischen Universität Darmstadt
zur Erlangung des akademischen Grades
eines Doktor-Ingenieurs (Dr.-Ing.)
genehmigte

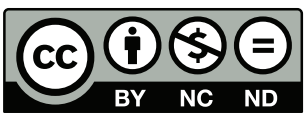
D i s s e r t a t i o n

von

Dipl.-Ing. Benedikt Klein
aus Mainz

Berichterstatter:	Prof. Dr.-Ing. habil. M. Oberlack
Mitberichterstatter:	Prof. Dr.-Ing. habil. J. Janicka
Tag der Einreichung:	30.06.2015
Tag der mündlichen Prüfung:	07.10.2015

Darmstadt, 2015
D17



Abstract

In this work, we present a high-order Discontinuous Galerkin Method (DGM) for simulating incompressible and variable density flows at low-Mach numbers. For steady cases, we apply the SIMPLE algorithm to solve the non-linear system in a segregated manner. For unsteady cases, the solver is implicit in time using backward differentiation formulae and the SIMPLE algorithm is applied to solve the non-linear system in each time step. The proposed method is implemented in the in-house software library BoSSS. The solver is extensively tested with respect to temporal and spatial convergence rates, performance and stability by simulating various test cases.

In the first part of this work, we describe the discretization and algorithm for incompressible flows. Using a mixed-order formulation for the spatial discretization, we obtain convergence rates of $k+1$ for velocity and k for pressure for various test cases, where k and $k-1$ are the orders of the approximation polynomials for velocity and pressure, respectively. Applying pressure stabilization for the equal-order formulation, the convergence rates are approximately the same, while the absolute error is smaller. By simulating the Orr-Sommerfeld problem we investigate the stability of the proposed method. The solver is validated by studying the two- and three-dimensional flow past a square cylinder. Main parts of this work concerning the solver and numerical results for incompressible flows have been published before by the author of this thesis in [KLEIN, B., KUMMER, F., OBERLACK, M. (2013): A SIMPLE based discontinuous Galerkin solver for steady incompressible flows. Journal of Computational Physics 237, 235–250] and [KLEIN, B., KUMMER, F., KEIL, M., OBERLACK, M. (2015): An extension of the SIMPLE based discontinuous Galerkin solver to unsteady incompressible flows. International Journal for Numerical Methods in Fluids 77, 10, 571–589].

In the second part of this work, the solver is extended to variable density flows at low-Mach numbers. An intermediate step in the development of the solver for low-Mach number flows is a method for simulating multiphase flows with a smooth interface approach and without surface tension. The solver for low-Mach number flows is based on the low-Mach number equations, which are an approximation of the compressible Navier-Stokes equations in the limit of zero Mach number. To the best of the author's knowledge, it is the first time that the DGM is applied to the low-Mach number equations. For spatial discretization the mixed-order formulation is applied. Various test cases confirm the high accuracy of the method also for multiphase flows and low-Mach number flows.

Zusammenfassung

Die vorliegende Arbeit beschäftigt sich mit einem numerischen Verfahren hoher Ordnung basierend auf der Diskontinuierlichen Galerkin Methode (DGM) zur Simulation inkompressibler Strömungen sowie Strömungen variabler Dichte bei kleinen Mach Zahlen. Die diskretisierten Gleichungen bilden ein nichtlineares Gleichungssystem, welche unter Verwendung des SIMPLE Algorithmus entkoppelt und iterativ gelöst werden. Dieses Verfahren wird sowohl für zeitunabhängige als auch zeitabhängige Probleme angewandt. Die Diskretisierung in der Zeit ist implizit, wobei sogenannte Rückwärtsdifferenzen verwendet werden. Die vorgeschlagene Methode wird in die institutseigene Softwarebibliothek BoSSS implementiert. Konvergenzraten in Zeit und Ort, Performance und Stabilität des entwickelten Lösers werden anhand zahlreicher Beispiele untersucht.

Im ersten Teil dieser Arbeit beschreiben wir die Diskretisierung sowie den Lösungsalgorithmus zur Simulation inkompressibler Strömungen. Für die örtliche Diskretisierung vergleichen wir die sogenannten mixed-order und equal-order Formulierungen. Bei der mixed-order Formulierung werden die Lösungen für die Geschwindigkeit durch Polynome der Ordnung k approximiert und für den Druck durch Polynome der Ordnung $k-1$. Bei der equal-order Formulierung besitzen alle Polynome die Ordnung k. Unter Verwendung der mixed-order Formulierung erhalten wir experimentelle Konvergenzraten von $k+1$ für die Geschwindigkeit und k für den Druck. Für die equal-order Formulierung ist eine Druckstabilisierung nötig. Die Konvergenzraten entsprechen näherungsweise jenen der mixed-order Formulierung, wobei die absoluten Fehler für die equal-order Formulierung kleiner sind. Die Stabilität des Verfahrens untersuchen wir anhand des Orr-Sommerfeld Problems. Schließlich validieren wir den Löser, indem wir die zwei- und dreidimensionale Zylinderumströmung untersuchen. Dieser Teil, welcher sich mit inkompressiblen Strömungen beschäftigt, basiert wesentlich auf zwei Artikeln, welche im Laufe dieser Arbeit entstanden sind: [KLEIN, B., KUMMER, F., OBERLACK, M. (2013): A SIMPLE based discontinuous Galerkin solver for steady incompressible flows. Journal of Computational Physics 237, 235–250] und [KLEIN, B., KUMMER, F., KEIL, M., OBERLACK, M. (2015): An extension of the SIMPLE based discontinuous Galerkin solver to unsteady incompressible flows. International Journal for Numerical Methods in Fluids 77, 10, 571–589].

Im zweiten Teil dieser Arbeit erweitern wir den Löser zur Simulation von Strömungen variabler Dichte bei kleinen Mach Zahlen. In einem Zwischenschritt entwickeln wir einen Löser zur Simulation von Mehrphasenströmungen. Dabei wird die physikalisch scharfe Grenzfläche zwischen zwei Phasen durch eine glatte Übergangsfunktion approximiert. Weiterhin wird die Oberflächenspannung vernachlässigt. Im nächsten Schritt wird die Simulation von sogenannten low-Mach Strömungen behandelt. Grundlage dieser Simulationen bilden die low-Mach Gleichungen, welche eine Approximation

der kompressiblen Navier-Stokes Gleichungen für kleine Mach Zahlen sind. Nach dem besten Wissen des Autors ist dies die erstmalige Anwendung der Diskontinuierlichen Galerkin Methode zur numerischen Lösung der low-Mach Gleichungen. Zur örtlichen Diskretisierung verwenden wir die mixed-order Formulierung. Verschiedene Testfälle demonstrieren die hohe Genauigkeit der Methode auch für Mehrphasenströmungen und low-Mach Strömungen.

Acknowledgements

This work was financially supported by the Center of Smart Interfaces through a seed fund project and in addition by the German Science Foundation (DFG) through Research Grant OB 96/31-1. Furthermore, some of the numerical results in this thesis have been obtained using the high performance computer Lichtenberg at the Technische Universität Darmstadt.

I would like to thank my supervisor Prof. Dr.-Ing. habil. Martin Oberlack for the opportunity to do this PhD thesis at the Chair of Fluid Dynamics and the trust and confidence he put into my work during the last four years. I also would like to thank Prof. Dr.-Ing. habil. Johannes Janicka for being my second supervisor and for supporting me in the early stages of my studies.

I am very thankful to Florian Kummer and Björn Müller for their support with the BoSSS code and their great passion they put into the code development, which built the basis for my work. With both of them I had very fruitful discussions on a regular basis throughout the whole period of this project. I also want to thank them for proofreading of this thesis and their suggestions for improvements. In addition, my thanks go to all current and former members of the BoSSS team. The weekly meetings we had and the countless discussions gave me a lot of input for that work. In particular, I want to mention (in alphabetical order) Nehzat Emamy, Christina Kallendorf, Stephan Krämer-Eis, Roozbeh Mousavi, Thomas Utz and Prof. Dr.-Ing. habil. Yongqi Wang for sharing their ideas.

I further acknowledge the work of Markus Keil, who contributed to the implementation of the solver for unsteady incompressible flows and to the results for the Taylor vortex flow and the two-dimensional flow past a square cylinder in his bachelor thesis (Keil 2012), as well as the work of Michaela Fischer, who performed the simulations for the three-dimensional flow past a square cylinder in her master thesis (Fischer 2014).

I am deeply grateful to all colleagues at the Chair of Fluid Dynamics. I appreciate the very pleasant working atmosphere and enjoyed being a member of this group.

Last but not least, I want to thank my whole family and all my friends for their continuous support and encouragement.

Contents

List of Figures

List of Tables

Nomenclature

$\mathcal{A}_{\alpha\alpha}$	Approximation matrix in SIMPLE algorithm
$\mathbf{A}_C$	Matrix for convection
$\mathbf{A}_{CT}$	Matrix for temperature advection
$\mathbf{A}_D$	Matrix for diffusion
$\mathbf{A}_{DT}$	Matrix for temperature diffusion
$\mathbf{A}_{ij}$	Matrix for variable density momentum equation
α_p	Relaxation factor for pressure
α_T	Relaxation factor for temperature
α_u	Relaxation factor for velocity
$\mathbf{A}_{P_i}$	Matrix for pressure gradient
$\underline{b}_B$	Summand for affine-linear offsets of continuity equation
$\underline{b}_{B_j}$	Affine-linear offset of divergence operator
$\underline{b}_{C_i}$	Affine-linear offset of convection operator
$\underline{b}_{CT}$	Affine-linear offset of temperature advection operator
$\underline{b}_{D_i}$	Affine-linear offset of diffusion operator
$\underline{b}_{CT}$	Affine-linear offset of temperature diffusion operator
$\underline{b}_i$	Summand for affine-linear offsets of momentum equation
$\mathbf{B}_j$	Matrix for divergence
$\underline{b}_{P_i}$	Affine-linear offset of pressure gradient operator
$\underline{b}_T$	Summand for affine-linear offsets of temperature equation
$\mathbf{C}$	Matrix for pressure stabilization
c	Generic scalar field
c_h	Discontinuous Galerkin approximation of scalar field c
$\overline{c_h^{\pm}}$	Mean value of $c_h^{\pm}$ in cell $K^{\pm}$
$\tilde{c}^{l,m}$	m-th polynomial coefficient of c_h in cell K^l
$\underline{\tilde{c}}$	Collection of all polynomial coefficients $\tilde{c}^{l,m}$
c_{K^l}	Expression for penalty parameter of symmetric interior penalty method
$\tilde{c}_p$	Specific heat capacity at constant pressure

$\mathbf{C}_T$	Matrix for temperature equation
d	Spatial dimension
ϵ_p	Convergence criterion for pressure
ϵ_T	Convergence criterion for temperature
ϵ_u	Convergence criterion for velocity
ϵ	Scaling factor for interface thickness
η	Penalty parameter of symmetric interior penalty method
$\hat{F}_{ij}^{C}$	Numerical flux function for convection
$\hat{F}_{j}^{CT}$	Numerical flux function for temperature advection
$\hat{F}_{i}^{D}$	Numerical flux function for diffusion
$\hat{F}_{j}^{\mathrm{Div}}$	Numerical flux function for divergence
$\hat{F}^{DT}$	Numerical flux function for temperature diffusion
f_i	Body force in i-th direction
$\hat{F}^{P}$	Numerical flux function for pressure gradient
Fr	Froude number
$\tilde{g}$	Constant of gravitation
Γ_I	Union of all interior faces of the numerical grid
h_E	Local length scale for pressure stabilization
$\mathbf{I}$	Identity matrix
k	Maximum polynomial order of elements of the space V_k
κ	Heat capacity ratio
K^l	l-th cell of the numerical grid
$\tilde{L}$	Non-dimensionless reference length
λ	Thermal conductivity
$\Lambda_{K,e}$	Parameter for local Lax-Friedrichs flux
$\tilde{\lambda}_\infty$	Non-dimensionless reference thermal conductivity
M	Mach number
m	Mass
$\mathbf{M}_\rho$	Density matrix
μ	Dynamic viscosity
$\mu_{\mathcal{A}}$	Dynamic viscosity of fluid $\mathcal{A}$ in two-phase flows
$\mu_{\mathcal{B}}$	Dynamic viscosity of fluid $\mathcal{B}$ in two-phase flows

$\tilde{\mu}_\infty$	Non-dimensionless reference dynamic viscosity
N	Number of grid cells
n	Superscript for time step
n_{ej}	Outward unit normal vector to ∂K^l
$n_j^\pm$	Outward unit normal vector to $\partial K^\pm$
N_k	Number of basis polynomials in each cell
Nu	Nusselt number
Ω	Computational domain
$\partial\Omega_D$	Domain boundary with Dirichlet condition
$\partial\Omega_N$	Domain boundary with outflow condition
$\partial\Omega_P$	Domain boundary with periodic condition
$\partial\Omega_{PO}$	Domain boundary with Dirichlet condition for pressure
p	Pressure
p_0	Hydrodynamic pressure in the low-Mach number approximation
p_2	Thermodynamic pressure in the low-Mach number approximation
φ	Level set function
P_k	Space of polynomials of degree at most k
$\mathbf{P}$	Matrix for pressure correction
Pr	Prandtl number
$\underline{\tilde{p}}^*$	Intermediate pressure in SIMPLE algorithm
$\underline{\tilde{p}}'$	Pressure correction in SIMPLE algorithm
$\mathbf{Q}(u_{hi})$	Flux Jacobian
$q^{l,m'}$	m'-th test / basis function in cell K^l (used for continuity equation)
Ra	Rayleigh number
Re	Reynolds number
$\mathcal{R}_h$	Residual of approximate solution c_h
ρ	Density
$\rho_\mathcal{A}$	Density of fluid $\mathcal{A}$ in two-phase flows
$\rho_\mathcal{B}$	Density of fluid $\mathcal{B}$ in two-phase flows
$\tilde{\rho}_\infty$	Non-dimensionless reference density
S	Material dependent constant in Sutherland's law
s	Order of BDF scheme
$\underline{\sigma}_\rho^{n+1}$	Summand for time steps of continuity equation

$\underline{\sigma}_T^n$	Summand for previous time steps of temperature equation
$\underline{\sigma}_{u_i}^n$	Summand for previous time steps of momentum equation
T	Temperature
t	Time
ϑ	Superscript for SIMPLE iterations
$\tilde{u}_\infty$	Non-dimensionless reference velocity
u_i	Velocity component
$\underline{\tilde{u}}_i^*$	Intermediate velocity in SIMPLE algorithm
$\underline{\tilde{u}}_i'$	Velocity correction in SIMPLE algorithm
$v^{l,m}$	m-th test / basis function in cell K^l (used for momentum and temperature equation)
V_k	Space of test and basis functions of maximum polynomial order k
x_i	Spatial coordinate in i-th direction

Abbreviations

BoSSS Bounded Support Spectral Solver . . . 1
CFD computational fluid dynamics . . . 1
DGM Discontinuous Galerkin Method . . . 1
DOF degrees of freedom . . . 1
FDM Finite Difference Method . . . 1
FEM Finite Element Method . . . 1
FVM Finite Volume Method . . . 1
PDE partial differential equation . . . 3
iIPSP intermediate language Parallel Scientific Platform . . . 12
LDG Local Discontinuous Galerkin . . . 13
NIP non-symmetric interior penalty . . . 13
SIMPLE Semi-Implicit Method for Pressure-Linked Equations . . . 13
SIP symmetric interior penalty . . . 13
ODE ordinary differential equation . . . 14
BDF backward differentiation formulae . . . 19
CG conjugate gradient . . . 33
GMRES generalized minimal residual . . . 33
DNS direct numerical simulation . . . 54
RANS Reynolds-averaged Navier-Stokes . . . 54

1 Introduction

In this thesis, we present a high-order Discontinuous Galerkin Method (DGM) for simulating incompressible and low-Mach number flows. Let Δx be a characteristic length of the numerical mesh. Then, the error e of a numerical method of order k behaves like $e \propto \Delta x^k$. In 2007, a survey within the computational fluid dynamics (CFD) community revealed that an order of three and above is considered to be high-order, cf. (Wang, Fidkowski, Abgrall, Bassi, Caraeni, Cary, Deconinck, Hartmann, Hillewaert, Huynh, Kroll, May, Persson, van Leer & Visbal 2013). State of the art CFD codes used in industry are based on the Finite Volume Method (FVM), which are usually of second order. In the field of research the high-order DGM has gained considerable interest for simulating flow problems, see the reviews for incompressible flows in Section 4.1 and for variable density flows at low-Mach numbers in Section 6.1.2. In general, the DGM can be of any order by choosing the order of the local polynomials, which are used to approximate the solution. The main motivation for using a high-order method is the capability to reach the same accuracy with less degrees of freedom (DOF) compared to a low-order method, or, in other words, to get a result of higher accuracy with the same number of DOF.

Besides the discretization technique, another main issue for any numerical method is the solution strategy which is used to solve the discretized equations. In this work, we apply the well-known SIMPLE algorithm, which was originally developed by Patankar & Spalding (1972) in the context of the Finite Difference Method (FDM) and is also extensively used in the context of the FVM, see e.g. (Ferziger & Perić 2002), and the Finite Element Method (FEM), cf. (Haroutunian, Engelman & Hasbani 1993). In the first part of this work we apply the SIMPLE based DGM to simulate steady as well as unsteady incompressible flows. Extensive numerical tests confirm the high order of the method and demonstrate the efficiency of the solution strategy. In the second part the developed solver is extended to variable density flows at low-Mach numbers. An intermediate step during the development of the low-Mach solver is a method for simulating multiphase flows applying a smooth interface approach. To the best of our knowledge it is the first time that the DGM is applied to solve the low-Mach number equations. Again, numerical tests demonstrate the high order and efficiency of the method.

The methods developed in this thesis have been implemented in the in-house software library Bounded Support Spectral Solver (BoSSS). All the numerical results shown herein have been simulated using BoSSS. The BoSSS code, which is based on the DGM, is currently under very active development at the Chair of Fluid Dynamics. The code development was started by Kummer, see (Kummer, Emamy, Mousavi Belfeh Teymouri & Oberlack 2009) and (Kummer 2012).

1.1 Outline of this work

The outline of the remainder of this thesis is as follows. In Chapter 2 we discuss the continuum mechanical models for incompressible flows, multiphase flows and low-Mach number flows. The basics of the Discontinuous Galerkin Method are described in Chapter 3. In this chapter also some definitions and notation are given, which are used throughout this work. The main features of the BoSSS code are also discussed in Chapter 3.

Main parts of Chapter 4 and Chapter 5 are based on (Klein, Kummer & Oberlack 2013) and (Klein, Kummer, Keil & Oberlack 2015), which have been published by the author of this thesis. In Chapter 4 we begin by reviewing Discontinuous Galerkin Methods for incompressible flows. Then, we describe the spatial as well as temporal discretization for incompressible flows. We also discuss the implementation of boundary conditions in a weak sense in the DGM. The SIMPLE algorithm to solve the discretized non-linear equations is also described in this chapter. Extensive numerical results for steady as well as unsteady incompressible flows are reported in Chapter 5. We assess the order of convergence of the method for different test cases with known analytical solutions. The performance of the implemented SIMPLE algorithm is examined in terms of SIMPLE iterations needed to reach a specified convergence criterion. By simulating the Orr-Sommerfeld problem we study the stability of the proposed method.

Chapter 6 and 7 are devoted to variable density flows. The former is subjected to the discretization and the SIMPLE algorithm for variable density flows and the latter to the numerical results. In Chapter 6 we review the literature applying the DGM to multiphase flows and low-Mach number flows. We also discuss the implementation of jump conditions for multiphase flows in the framework of the DGM. In Chapter 7 we verify the implementation of the variable density solver by simulating test cases for multiphase flows as well as low-Mach number flows. Like for the incompressible case we investigate the spatial accuracy as well as the performance of the SIMPLE algorithm. Finally, in Chapter 8 we give some conclusions and an outlook.

2 Continuum mechanical models

We begin in this chapter by describing the continuum mechanical models, i.e. the partial differential equations (PDEs) and algebraic relations, for incompressible flows in Section 2.1, for multiphase flows in Section 2.2 and for low-Mach number flows in Section 2.3. These PDEs will build the basis for the discretization, algorithms and numerical results in the remainder of this thesis.

2.1 Incompressible flows

For incompressible flows we consider the Navier-Stokes equations in the following dimensionless form

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{2.1a}$$

$$\frac{\partial u_i}{\partial t} + \frac{\partial (u_i u_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + f_i, \tag{2.1b}$$

where u_i is the i-th component of the velocity vector, p is the pressure and f_i is any known body force. In (2.1a)-(2.1b) and throughout this work we use the Einstein summation convention, which implies summation over terms with indices which appear twice. The indices i and j vary from 1 to d, where $d \in \{2, 3\}$ is the dimension of the computational domain Ω. There is only one dimensionless parameter in the incompressible Navier-Stokes equations, namely the Reynolds number, which is defined as

$$Re = \frac{\tilde{\rho}_\infty \tilde{u}_\infty \tilde{L}}{\tilde{\mu}_\infty}, \tag{2.2}$$

where $\tilde{\rho}_\infty$, $\tilde{u}_\infty$, $\tilde{L}$ and $\tilde{\mu}_\infty$ are non-dimensionless reference values for density, velocity, length scale and dynamic viscosity, respectively. The set of PDEs is supplemented by initial conditions for the velocity

$$u_i(t = 0) = u_i^0 \quad \text{in } \Omega, \tag{2.3}$$

and the following boundary conditions

$$\partial\Omega_D : \quad u_i = u_{Di}, \tag{2.4a}$$

$$\partial\Omega_N : \quad \frac{1}{Re}\frac{\partial u_i}{\partial n} - p n_i = 0, \tag{2.4b}$$

$$\partial\Omega_{PO} : p = p_D, \tag{2.4c}$$

$$\partial\Omega_P: \quad u_i(x_j) = u_i(x'_j),\ p(x_j) = p(x'_j). \tag{2.4d}$$

At inlet parts of the domain boundary and at walls, Dirichlet values for the velocity have to be prescribed, cf. (2.4a). At outlets either the outflow condition (2.4b) can be used or a Dirichlet value for the pressure can be given (2.4c). Note that (2.4b) is a common outflow boundary condition in the FEM community, which preserves the symmetry of the discrete gradient and divergence operators. For more details on that issue we refer the reader to (Gresho 1991). The implementation of the boundary conditions will be discussed in Section 4.4, showing that (2.4b) is also a natural choice for the Discontinuous Galerkin discretization. Finally, by (2.4d) periodic boundary conditions are represented with x_j and x'_j being corresponding periodic points.

2.2 Multiphase flows

Within the scope of this work, by multiphase flow we refer to the flow of two immiscible and incompressible fluids $\mathcal{A}$ and $\mathcal{B}$ with distinct density and viscosity, which are separated by a moving interface. We consider only material interfaces, i.e. the velocity in the normal direction of both faces at the interface and the interface velocity itself are identical. The position of the interface is defined by the 0.5 iso-contour of the level set function $\varphi \in [0,1]$, which is advected with the velocity of the flow field

$$\frac{\partial \varphi}{\partial t} + u_j \frac{\partial \varphi}{\partial x_j} = 0. \tag{2.5}$$

The level set function φ varies smoothly from one in phase $\mathcal{A}$ to zero in phase $\mathcal{B}$, i.e. the interface is of finite thickness and density and viscosity are smeared out over a transition layer

$$\rho = \rho_{\mathcal{A}}\varphi + \rho_{\mathcal{B}}(1-\varphi), \tag{2.6a}$$
$$\mu = \mu_{\mathcal{A}}\varphi + \mu_{\mathcal{B}}(1-\varphi), \tag{2.6b}$$

where the constants $\rho_{\mathcal{A}}$, $\rho_{\mathcal{B}}$ and $\mu_{\mathcal{A}}$, $\mu_{\mathcal{B}}$ are the density and viscosity of the corresponding phase. In contrast one could use sharp interface models, which are numerically more difficult and subject of ongoing research, cf. (Heimann, Engwer, Ippisch & Bastian 2013), (Kummer & Oberlack 2013) and (Kummer 2013). Our motivation here to apply a smooth interface approach is, that the equations derived below are an almost natural intermediate step going from incompressible to low-Mach number flows. Hence, these kinds of flows are very useful for verification in the process of code development. And still, smooth interface models based on the level set function can be considered to be state of the art. For example (Pochet, Hillewaert, Geuzaine, Remacle & Marchandise 2013) propose a DGM using a signed distance level set function, where smoothing of density and viscosity is done by a hyperbolic tangent function of the level set. The approach we apply, where the level set itself is a smooth function, was

developed in the context of the FEM in (Olsson & Kreiss 2005) and (Olsson, Kreiss & Zahedi 2007).

Next, we describe the continuity and momentum equations for multiphase flows with a smooth interface. Even though the density is not constant in the transition layer, the velocity field is still divergence-free, which can be found by looking at the material derivative of the density

$$\frac{D\rho}{Dt} = \frac{\partial \rho}{\partial t} + u_j \frac{\partial \rho}{\partial x_j} = \frac{d\rho}{d\varphi}\frac{\partial \varphi}{\partial t} + u_j \frac{d\rho}{d\varphi}\frac{\partial \varphi}{\partial x_j} = \frac{d\rho}{d\varphi} \underbrace{\left(\frac{\partial \varphi}{\partial t} + u_j \frac{\partial \varphi}{\partial x_j} \right)}_{\overset{(2.5)}{=} 0} = 0. \tag{2.7}$$

Using this result the full set of PDEs describing multiphase flows with a smooth interface can be cast in the following form

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{2.8a}$$

$$\frac{\partial \rho u_i}{\partial t} + \frac{\partial \rho u_i u_j}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\mu \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \right) + f_i, \tag{2.8b}$$

$$\frac{\partial \varphi}{\partial t} + \frac{\partial u_j \varphi}{\partial x_j} = 0. \tag{2.8c}$$

In (2.8c) we have used that the velocity field is divergence-free to write the equation for the level set advection (2.5) in conservative form. In the momentum equation (2.8b) surface tension is not taken into account. In general, one needs to assure that the interface thickness stays constant over time. This can be accomplished be either solving a reinitialization equation or by modifying the transport equation for the level set function (2.8c), cf. (Olsson & Kreiss 2005) and (Olsson et al. 2007). For the test cases presented in this thesis no such treatment is necessary.

Initial conditions need to be given for the velocity and the level set, i.e.

$$u_i(t=0) = u_i^0 \quad \text{in } \Omega, \tag{2.9a}$$

$$\varphi(t=0) = \varphi^0 \quad \text{in } \Omega. \tag{2.9b}$$

The level set needs to be initialized according to the initial position of the interface and the interface thickness. For one-dimensional test cases the level set can e.g. be initialized by

$$\varphi(x_1, t=0) = \frac{1}{2} - \frac{1}{2}\tanh\left(\frac{\pi(x_1 - x_{\mathcal{I}}(t=0))}{\epsilon} \right), \tag{2.10}$$

where $x_{\mathcal{I}}(t=0)$ is the initial position of the interface and ϵ scales the thickness of the interface. For the initialization of two- and three-dimensional test cases we refer the

reader to (Olsson & Kreiss 2005). Finally, the model is completed by the boundary conditions

$$\partial\Omega_D: \quad u_i = u_{Di},\ \varphi = \varphi_D, \tag{2.11a}$$
$$\partial\Omega_{DN}: \ u_i = u_{Di}, \tag{2.11b}$$
$$\partial\Omega_N: \quad \frac{\mu}{Re}\frac{\partial u_i}{\partial n} - pn_i = 0, \tag{2.11c}$$
$$\partial\Omega_{PO}: \ p = p_D, \tag{2.11d}$$
$$\partial\Omega_P: \quad u_i(x_j) = u_i(x_j'),\ p(x_j) = p(x_j'),\ \varphi(x_j) = \varphi(x_j'). \tag{2.11e}$$

At $\partial\Omega_{DN}$ Dirichlet values for the velocity are set, but no boundary condition for the level set function is prescribed, e.g. at a wall. The remaining boundary conditions are the counterparts for the conditions which already have been discussed for incompressible flows in the previous section.

2.3 Low-Mach number flows

The low-Mach number equations are an approximation for flows where the density changes due to temperature variations, but is not influenced by the hydrodynamic pressure, i.e. acoustic affects are neglected. Such flows occur in many technical applications, like e.g. combustion processes, heating and cooling devices and in natural convection flows. All these flows have in common that the velocity of the flow field is small compared to the speed of sound, i.e. the Mach number

$$M = \frac{\tilde{u}_\infty}{\sqrt{\kappa \tilde{p}_\infty / \tilde{\rho}_\infty}}, \tag{2.12}$$

is small. In (2.12) κ is the heat capacity ratio. Detailed derivations of the low-Mach number equations can e.g. be found in (Majda & Sethian 1985), (Rook 2001) and (Rauwoens, Vierendeels, Dick & Merci 2009). Here, we briefly sketch the principal ideas and comment on the main consequences. The equations are derived starting from the dimensionless compressible Navier-Stokes equations. Each variable is expanded into a series of the Mach number, e.g. the pressure is written as

$$p = p_0 + Mp_1 + \kappa M^2 p_2 + \mathcal{O}(M^3). \tag{2.13}$$

The low-Mach number equations are obtained by applying the series expansion (2.13) to all variables and taking the limit of zero Mach number, viz.

$$p_0 = p_0(t), \tag{2.14a}$$
$$\frac{\partial \rho}{\partial t} + \frac{\partial(\rho u_j)}{\partial x_j} = 0, \tag{2.14b}$$

$$\frac{\partial(\rho u_i)}{\partial t} + \frac{\partial(\rho u_i u_j)}{\partial x_j} = -\frac{\partial p_2}{\partial x_i} + \frac{1}{Re}\frac{\partial}{\partial x_j}\left(\mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\mu\delta_{ij}\frac{\partial u_k}{\partial x_k}\right) - \frac{1}{Fr^2}\rho\delta_{i2}, \tag{2.14c}$$

$$\frac{1}{\kappa}\frac{\partial(\rho T)}{\partial t} + \frac{\partial(\rho u_j T)}{\partial x_j} = \frac{1}{RePr}\frac{\partial}{\partial x_j}\left(\lambda\frac{\partial T}{\partial x_j}\right). \tag{2.14d}$$

The equation of state, assuming ideal gas, takes the following form for M going to zero

$$\rho = \frac{p_0}{T}, \tag{2.15}$$

and the Sutherland's law for viscosity becomes

$$\mu = T^{\frac{3}{2}}\frac{1 + \frac{S}{\tilde{T}_\infty}}{T + \frac{S}{\tilde{T}_\infty}}, \tag{2.16}$$

where S is a material dependent constant[1] and $\tilde{T}_\infty$ is a non-dimensionless reference temperature. Assuming constant Prandtl number, the dimensionless thermal conductivity is given by

$$\lambda = \mu. \tag{2.17}$$

In (2.14a)-(2.16) the expansion index has been dropped for all variables except for the pressure, since two parts of the pressure remain in the zero Mach number limit. The first part p_0 is constant in space and called thermodynamic pressure, as p_0 enters in the equation of state (2.15) and determines the density. The second part p_2 is referred to as hydrodynamic pressure, since it occurs in the momentum equation (2.14c). There are three dimensionless parameters in the low-Mach number equations, viz. the Reynolds, Froude and Prandtl numbers

$$Re = \frac{\tilde{\rho}_\infty \tilde{u}_\infty \tilde{L}}{\tilde{\mu}_\infty}, \tag{2.18a}$$

$$Fr = \frac{\tilde{u}_\infty}{\sqrt{\tilde{g}\tilde{L}}}, \tag{2.18b}$$

$$Pr = \frac{\tilde{c_p}\tilde{\mu}_\infty}{\tilde{\lambda}_\infty}, \tag{2.18c}$$

where compared to incompressible flows additional non-dimensionless reference values have to be introduced for the specific heat capacity $\tilde{c}_p$, the thermal conductivity $\tilde{\lambda}_\infty$ and the constant of gravitation $\tilde{g}$.

For low-Mach number flows we apply the following boundary conditions

$$\partial\Omega_D: \quad u_i = u_{Di},\ T = T_D, \tag{2.19a}$$

[1] All simulations in this work have been performed assuming air, where $S = 110.5K$.

$$\partial\Omega_{DN} : u_i = u_{Di},\ \frac{\partial T}{\partial n} = 0, \tag{2.19b}$$

$$\partial\Omega_N : \ \frac{\mu}{Re}\frac{\partial u_i}{\partial n} - pn_i = 0,\ \frac{\partial T}{\partial n} = 0, \tag{2.19c}$$

$$\partial\Omega_{PO} : \ p = p_D,\ \frac{\partial T}{\partial n} = 0, \tag{2.19d}$$

$$\partial\Omega_P : \ \ u_i(x_j) = u_i(x_j'),\ p(x_j) = p(x_j'),\ T(x_j) = T(x_j'). \tag{2.19e}$$

There are two different types of boundary conditions, $\partial\Omega_D$ and $\partial\Omega_{DN}$, for walls. The first one is a general Dirichlet boundary condition, which is also used at an inlet, where values for the velocity and the temperature are prescribed. The second one is used at adiabatic walls, where the velocity is set to a given value, but the heat transfer through the wall should be zero. At an outflow $\partial\Omega_N$ and $\partial\Omega_{PO}$ the boundary conditions for the flow field are the same like for incompressible flows and the gradient of the temperature in the normal direction is set to zero. By (2.19e) periodic boundary conditions are defined on $\partial\Omega_P$.

Initial conditions have to be given for the velocity and the temperature

$$u_i(t = 0) = u_i^0 \quad \text{in } \Omega, \tag{2.20a}$$

$$T(t = 0) = T^0 \quad \text{in } \Omega. \tag{2.20b}$$

Special attention has to be paid to the calculation of the thermodynamic pressure p_0. As already mentioned above p_0 is constant in space and only a function of time, cf. (2.14a). Therefore, in open systems p_0 is also assumed to be constant in time and hence the thermodynamic pressure is a parameter, which has to be set prior to the simulations. In closed systems p_0 has to be determined to guarantee mass conservation. Given the initial mass in the system $m(t = 0)$, p_0 can be calculated using the equation of state (2.15) for ideal gas, that is

$$p_0(t) = \frac{m(t = 0)}{\int \frac{1}{T} dV}. \tag{2.21}$$

3 The Discontinuous Galerkin Method

Within about the last 10 years the DGM has gained considerable interest for the simulation of incompressible and variable density flows, cf. the reviews for incompressible flows in Section 4.1 and for variable density flows in Section 6.1. In this chapter we introduce the DGM for a simple model problem, namely the linear advection equation, and comment on the main properties of the DGM. For further reading about the DGM we refer to the text books (Hesthaven & Warburton 2007) and (Li 2006) as well as the essay (Cockburn 2003).

The DGM has got several valuable properties. First of all, the DGM is of high-order accuracy. Assuming a characteristic mesh size h and a polynomial order k in each cell of the numerical grid, a convergence rate for the spatial error of $\mathcal{O}(h^{k+1/2})$ can be proven for a scalar hyperbolic equation (Johnson & Pitkäranta 1986). For many applications one observes experimental convergence rates of $\mathcal{O}(h^{k+1})$, see also the numerical results in this work in Chapter 5 for incompressible flows and, respectively, in Chapter 7 for variable density flows. Independently of the order k, each cell of the numerical grid is only coupled to its immediate neighbors, which is ideally suited for parallelization. Furthermore, the DGM can easily handle complex geometries with hanging nodes and is locally conservative.

3.1 Basics

We introduce the Discontinuous Galerkin Method by considering the linear advection equation for some smooth scalar field c given by

$$\frac{\partial c}{\partial t} + \frac{\partial (u_j c)}{\partial x_j} = 0, \tag{3.1}$$

where u_j is a prescribed velocity field. Tesselating the computational domain Ω into N simplicial elements K^l, we seek an approximate solution c_h, which is defined locally in each cell K^l using a modal basis

$$c(x_j,t)|_{K^l} \approx c_h(x_j,t)|_{K^l} = \sum_{m=1}^{N_k} \tilde{c}^{l,m}(t) v^{l,m}(x_j). \tag{3.2}$$

In (3.2), $\tilde{c}^{l,m}(t)$ are the unknown polynomial coefficients, $v^{l,m}(x_j)$ are the basis polynomials of maximum order k and N_k is the corresponding number of basis polynomials in each cell. Inserting the approximation (3.2) in (3.1), we can define the residual

$$\mathcal{R}_h(x_i,t) = \frac{\partial c_h}{\partial t} + \frac{\partial u_j c_h}{\partial x_j}. \tag{3.3}$$

Then, we require for the approximate solution c_h that the residual vanishes in a weak sense on all elements K^l

$$\int_{K^l} \mathcal{R}_h(x_i,t) v^{l,m}(x_j)\, dx = 0, \quad 1 \leq m \leq N_k, \tag{3.4}$$

where the test functions are identical to the basis functions leading to a Galerkin scheme. Using the definition of the residual (3.3) and carrying out integration by parts yields the final weak form of the Discontinuous Galerkin Method

$$\int_{K^l} \frac{\partial c_h}{\partial t} v^{l,m}\, dx + \int_{\partial K^l} \widehat{u_j c_h} v^{l,m} n_{ej}\, ds - \int_{K^l} u_j c_h \frac{\partial v^{l,m}}{\partial x_j}\, dx = 0, \quad 1 \leq m \leq N_k, \tag{3.5}$$

where n_{ej} is the outward unit normal vector to ∂K^l. In (3.5) we already introduced the numerical flux $\widehat{u_j c_h}$ in the surface integral. For the Discontinuous Galerkin Method there is no restriction on the continuity of the solution at the element face $e = \overline{K^+} \cap \overline{K^-}$ between two adjacent elements K^+ and K^-. Hence, the surface integral in (3.5) is not uniquely defined. The test function $v^{l,m}$ is evaluated using the inner values in cell K^l. The coupling of the solution between K^+ and K^- is realized by the numerical flux $\widehat{u_j c_h}$, which will be a function of the traces on the element face e. Note that through the numerical flux each cell is only coupled to its immediate neighbors, i.e. the first layer of neighbor cells, independently of the polynomial order k. This renders the DGM perfectly well suited for parallelization. Another favorable property of the DGM is the block diagonal structure of the mass matrix as a result of the local basis and test functions. In fact, using orthonormal basis functions, i.e.

$$\int_{K^l} v^{l,m} v^{l,n}\, dx = \delta_{m,n}, \tag{3.6}$$

the mass matrix reduces to the identity matrix, which can be easily found by inserting the approximation (3.2) in the first volume integral of (3.5) and applying (3.6). Next, we take a look at the conservativity of the DGM. By inserting the test function of zeroth order, i.e. a constant function, in the weak form of the DGM (3.5) we obtain

$$\int_{K^l} \frac{\partial c_h}{\partial t}\, dx + \int_{\partial K^l} \widehat{u_j c_h} n_{ej}\, ds = 0, \tag{3.7}$$

which shows the local conservativity of the DGM. The conservativity of a numerical method is highly appreciated by the CFD community. Since the constant function will be always contained in the space of test functions, (3.7) is true for any DGM of arbitrary order k.

To sum up, the central part of the DGM is the definition of the numerical fluxes, which affect the consistency, stability as well as accuracy, cf. (Cockburn 2003). In addition, the boundary conditions of the PDEs are imposed weakly by the evaluation of the numerical fluxes at the domain boundary. The fluxes used in this work and the implementation of the boundary conditions for incompressible and variable density flows are discussed in Chapters 4 and 6, respectively. Before, we need to introduce some definitions and notation in the next section.

3.2 Definitions and notation

The space of test and basis functions is defined to be

$$V_k := \{v \in L^2(\Omega) |\ v|_{K^l} \in P_k(K^l)\ \forall l = 1, \dots, N\}, \tag{3.8}$$

where $P_k(K^l)$ is the space of polynomials of degree at most k. By the vector $\underline{\tilde{c}}$ we denote a collection of all DG coefficients $\tilde{c}^{l,m}$. Let (l,m) be at two-dimensional multi-index. Then, the (l,m)-th component of $\underline{\tilde{c}}$ is defined by $\underline{\tilde{c}}_{(l,m)} := \tilde{c}^{l,m}$.

By Γ_I we denote the union of all interior faces $e = \overline{K^+} \cap \overline{K^-}$. The numerical fluxes will be functions of the traces on $e \subseteq \Gamma_I$, which are defined as follows

$$c_h^{\pm} = \lim_{\epsilon \downarrow 0} c_h(x_j - \epsilon n_j^{\pm}), \tag{3.9a}$$

where $n_j^{\pm}$ are the outward unit normal vectors to $\partial K^{\pm}$. On boundary faces $e \subseteq \partial\Omega$, we will denote the inner values by c_h^+, while c_h^- remains undefined on $e \subseteq \partial\Omega$. Next, we introduce the following jump and average operators on inner edges $e \subseteq \Gamma_I$

$$[\![c_h]\!] = c_h^+ n_j^+ + c_h^- n_j^-, \tag{3.10a}$$

$$\{c_h\} = \frac{1}{2}\left(c_h^+ + c_h^-\right). \tag{3.10b}$$

The evaluation of the jump and average operators at the boundary edges will be given separately for each numerical flux, see Chapter 4 for incompressible flows and Chapter 6 for variable density flows.

3.3 The software library BoSSS

The development of the software library BoSSS was started by Kummer, see (Kummer et al. 2009) and (Kummer 2012). BoSSS is a framework for the numerical solution of arbitrary conservation laws based on the Discontinuous Galerkin Method. Different conservation laws can easily be implemented based on a very generic interface by defining the volume and surface integrals of the weak form of the DGM (3.5). The library is programmed in an object-oriented design in C# and MPI is used for parallelization. Various linear grid types (triangles, quads, tetras, cubes) as well as curved

elements can be used in all dimensions (1D-3D). The structure of the BoSSS framework is basically divided into six layers, cf. (Kummer 2012):

- Native: Third-party code (e.g. MPI, ParMETIS, Hypre, Pardiso)
- Layer 0 (intermediate language Parallel Scientific Platform (ilPSP)): Sparse matrices and vectors, wrappers for third-party code
- Layer 1 (Platform): Utilities (e.g. 2D and 3D vectors, block-diagonal matrices)
- Layer 2 (Foundation): Grid handling, quadrature, input/output
- Layer 3 (Solution): Spatial discretization (weak form) for physical applications of layer 4, tools for postprocessing (e.g. plotting, error evaluation), parsing of control file
- Layer 4 (Application): Solvers for various physical applications, e.g. compressible flows, incompressible flows, level set methods for multiphase flows, low-Mach number flows.

On layer 4 various physical applications have been implemented so far. A solver for the compressible Euler equations was implemented by Müller (2014). Emamy (2014) has developed a multiphase solver with different dielectric permittivity exposed to a stationary electric field. The solution of the level set advection equation and its re-initialization has been treated by Mousavi (2014).

In this work the development of a segregated implicit solver for the steady incompressible Navier-Stokes equations is continued, which was started by the author in (Klein 2011). The steady solver was verified by simulating various test cases, which have been published in (Klein et al. 2013). The solver is based on the well-known SIMPLE algorithm, which was developed in the context of the FDM by Patankar & Spalding (1972). In the course of this thesis the solver was first extended to unsteady incompressible flows. Preliminary results have been obtained by Keil (2012). Later the unsteady solver has been extensively verified and the results have been published in (Klein et al. 2015). The solver was also used to simulate the unsteady three-dimensional flow around a square cylinder by Fischer (2014) and the Taylor-Couette flow applying curved elements by Utz (2014). Then, the solver was further extended to variable density flows. The implementation of the SIMPLE based Discontinuous Galerkin solver for incompressible and low-Mach number flows is concerned with layer 3 of the BoSSS framework, where the weak form of the governing equations is implemented, and layer 4, where the SIMPLE algorithm is implemented.

4 Discretization and algorithm for incompressible flows

In this chapter, we discuss the discretization of the incompressible Navier-Stokes equations by the DGM as well as the algorithm to solve the non-linear discrete system of equations. Main parts of this chapter are based on the two publications (Klein et al. 2013) and (Klein et al. 2015).

We begin in Section 4.1 [1] by reviewing the literature on the application of the DGM to the incompressible Navier-Stokes equations and motivate our approach to adapt the Semi-Implicit Method for Pressure-Linked Equations (SIMPLE) algorithm. In Section 4.2 [2] we discuss the spatial discretization, which is followed by the temporal discretization in Section 4.3 [3]. Then, in Section 4.4 we comment on the implementation of boundary conditions. Finally, we discuss the SIMPLE algorithm in Section 4.5 [4].

4.1 Review of Discontinuous Galerkin solvers

Within recent years, the application of the DGM to the steady and unsteady incompressible Navier-Stokes equations has been reported by several authors. One of the first schemes was the Local Discontinuous Galerkin (LDG) Method, which was developed by Cockburn, Kanschat & Schötzau (2005). The name LDG stems from the discretization of the viscous term. In (Cockburn, Kanschat & Schötzau 2009) this approach was extended to equal-order formulations, which was stabilized by adding a penalty term for pressure jumps to the continuity equation, cf. Section 4.2.2.4. A fixed point iteration is used to solve the non-linear system in a fully coupled manner for steady cases. In (Girault, Rivière & Wheeler 2005) the symmetric interior penalty (SIP) and non-symmetric interior penalty (NIP) method are compared for the discretization of the viscous term. A differential splitting technique is used for temporal discretization and decoupling of the equations. Rivière & Girault (2006) also compare the SIP and NIP and use Picard iterations for solving the steady equations. In (Bassi, Crivellini, Di Pietro & Rebay 2006) and (Bassi, Crivellini, Di Pietro & Rebay 2007) an artificial compressibility flux is developed for the transport term and time-marching is done by implicit Runge-Kutta schemes. Algebraic and differential splitting techniques are used in the schemes proposed by Shahbazi, Fischer & Ethier (2007) and Ferrer & Willden (2011), respectively. In both schemes the transport term is treated explicitly and the viscous term is treated implicitly, which is discretized by the SIP method. In

[1] Modified version of (Klein et al. 2013, Sect. 1).
[2] Modified version of (Klein et al. 2013, Sect. 2) and (Klein et al. 2015, Sect. 2.1).
[3] Taken from (Klein et al. 2015, Sect. 2.2).
[4] Taken from (Klein et al. 2015, Sect. 3), added new Section 4.5.1 on pressure reference point.

(Rhebergen, Cockburn & van der Vegt 2013) the DGM is used for space as well as time discretization leading to a so-called space-time Discontinuous Galerkin Method. The method is implicit in time and again Picard iterations are used for the non-linear system. Tavelli & Dumbser (2014) propose a semi-implicit DGM for incompressible flows on staggered grids, which is stable for equal-order formulations.

Independently of the DG discretization, there are some difficulties when simulating incompressible flows: (i) the equations are non-linear due to the convective part; (ii) velocity and pressure are strongly coupled via an elliptic operator; (iii) an explicit equation to calculate the pressure is missing. The solution techniques found in the context of the DGM in the references mentioned above can be divided into three groups:

(a) *fixed point iterations*: solving the non-linear system for all velocity components and the pressure in a fully coupled manner;

(b) *operator splitting techniques*: for unsteady cases, either at the algebraic or differential level;

(c) *implicit Runge-Kutta schemes*: all velocity components and the pressure are solved in a fully coupled manner.

Apart from these solution techniques, in 1972, the well-known SIMPLE algorithm was proposed by Patankar & Spalding (1972) in the context of the FDM. This method has proved itself in many cases to be very efficient for simulating incompressible flows and is used toady in almost every FEM / FVM CFD Code. The SIMPLE algorithm has got several unique features compared to the techniques described above in the groups (a)-(c). It can be applied for solving steady state problems without the need of using any time derivative. The solution schemes of group (b) and (c) are inherently unsteady and need to solve the time-dependent equations even for stationary cases until a steady state is reached. The core part of the SIMPLE algorithm is the introduction of an iterative process such that the discrete equations get linearized and decoupled in each velocity component and the pressure. The techniques of group (a) and (c) always solve all unknowns, i.e. all velocity components and the pressure, in a fully coupled manner resulting in a much larger system of equations. An equation for the pressure is derived on the discrete level. For unsteady problems fully implicit time schemes can be employed, i.e. the time step size is not restricted by the CFL condition. In the solution schemes of group (b) the convective part is usually treated explicitly leading to a restriction in the size of the time step by the CFL condition. Therefore, we propose a method, which adapts the SIMPLE algorithm to solve the incompressible Navier-Stokes equations discretized by the DGM.

4.2 Spatial discretization

In this section, we first derive the semidiscrete system of equations, i.e. a system of ordinary differential equations (ODEs) in time, which has been discretized in space.

Then, we give the definitions for the numerical fluxes occurring in the weak form of the momentum and continuity equation.

4.2.1 Semidiscrete system of equations

We seek the numerical solution of the unsteady incompressible Navier-Stokes equations (2.1a) and (2.1b) with the initial conditions (2.3) and the boundary conditions (2.4a)-(2.4d) for the unknown DG coefficients $\tilde{u}_i^{l,m}$ and $\tilde{p}^{l,m'}$ of the approximated velocity $u_{hi} \in V_k$ and the approximated pressure $p_h \in V_{k'}$, where $k' = k-1$ for the mixed-order formulation and $k' = k$ for the equal-order formulation. The spatial discretization of (2.1a)-(2.1b) is derived by multiplying with a test function $v^{l,m} \in V_k$ for the operators of the momentum equation (2.1b) and $q^{l,m'} \in V_{k'}$ for the divergence operator of the continuity equation (2.1a), replacing u_i and p by their DG approximations u_{hi} and p_h and integrating over an element K^l

$$\left\langle \frac{\partial u_{hi}}{\partial t}, v^{l,m} \right\rangle = \int_{K^l} \frac{\partial u_{hi}}{\partial t} v^{l,m} d\mathbf{x} = \left[\frac{\partial \underline{\tilde{u}}_i}{\partial t} \mathbf{I} \right]_{(l,m)}, \tag{4.1a}$$

$$\left\langle \frac{\partial u_{hi} u_{hj}}{\partial x_j}, v^{l,m} \right\rangle \approx \int_{\partial K^l} \hat{F}_{ij}^{C} n_{ej} v^{l,m} ds - \int_{K^l} u_{hi} u_{hj} \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_C(\underline{\tilde{u}}) \underline{\tilde{u}}_i + \underline{b}_{C_i} \right]_{(l,m)}, \tag{4.1b}$$

$$\left\langle \frac{\partial^2 u_{hi}}{\partial x_j \partial x_j}, v^{l,m} \right\rangle \approx \int_{\partial K^l} \hat{F}_i^{D} ds - \int_{K^l} \frac{\partial u_{hi}}{\partial x_j} \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_D \underline{\tilde{u}}_i + \underline{b}_{D_i} \right]_{(l,m)}, \tag{4.1c}$$

$$\left\langle \frac{\partial p_h}{\partial x_i}, v^{l,m} \right\rangle \approx \int_{\partial K^l} \hat{F}^{P} n_{ei} v^{l,m} ds - \int_{K^l} p_h \frac{\partial v^{l,m}}{\partial x_i} d\mathbf{x} =: \left[\mathbf{A}_{P_i} \underline{\tilde{p}} + \underline{b}_{P_i} \right]_{(l,m)}, \tag{4.1d}$$

$$\left\langle \frac{\partial u_{hj}}{\partial x_j}, q^{l,m'} \right\rangle \approx \int_{\partial K^l} \hat{F}_j^{\mathrm{Div}} n_{ej} q^{l,m'} ds - \int_{K^l} u_{hj} \frac{\partial q^{l,m'}}{\partial x_j} d\mathbf{x} =: \left[\sum_{j=1}^{d} \left(\mathbf{B}_j \underline{\tilde{u}}_j + \underline{b}_{B_j} \right) + \mathbf{C} \underline{\tilde{p}} \right]_{(l,m')}, \tag{4.1e}$$

$$\left\langle -f_i, v^{l,m} \right\rangle = \int_{K^l} -f_i v^{l,m} d\mathbf{x} =: \left[\underline{b}_{f_i} \right]_{(l,m)}. \tag{4.1f}$$

Note that the test functions $v^{l,m}$ and $q^{l,m'}$ in the edge integrals of (4.1a)-(4.1f) are evaluated using the inner values in cell K^l. In (4.1a) we obtain the identity matrix for the mass matrix due to the orthonormality of the basis functions. The numerical fluxes $\hat{F}_{ij}^{C}$, $\hat{F}_i^{D}$, $\hat{F}^{P}$ and $\hat{F}_j^{\mathrm{Div}}$ in (4.1b)-(4.1e) for the definition of the discrete operators $\mathbf{A}_C(\underline{\tilde{u}})$, $\mathbf{A}_D$, $\mathbf{A}_{P_i}$, $\mathbf{B}_j$ and $\mathbf{C}$ are given below. The non-linear term in (4.1b) is written as $\mathbf{A}_C(\underline{\tilde{u}})\underline{\tilde{u}}_i$, where $\underline{\tilde{u}}$ shall denote a vector containing all DG coefficients of all spatial velocity components. Within the SIMPLE algorithm, this term will be linearized, cf.

Section 4.5. The vectors $\underline{b}_{C_i}$, $\underline{b}_{D_i}$, $\underline{b}_{P_i}$, $\underline{b}_{B_j}$ are the affine-linear offsets of the operators following from non-zero boundary conditions and $\underline{b}_{f_i}$ includes all source terms. Setting

$$\underline{b}_i := -\underline{b}_{C_i} + \underline{b}_{D_i} - \underline{b}_{P_i} - \underline{b}_{f_i} \quad \text{and} \quad \underline{b}_B := -\sum_{j=1}^{d} \underline{b}_{B_j}, \tag{4.2}$$

we can write the semidiscrete system of equations

$$\frac{\partial \underline{\tilde{u}}_i}{\partial t} + \left(\mathbf{A}_C(\underline{\tilde{u}}) - \frac{1}{Re}\mathbf{A}_D \right) \underline{\tilde{u}}_i + \mathbf{A}_{P_i} \underline{\tilde{p}} = \underline{b}_i, \tag{4.3a}$$

$$\mathbf{B}_j \underline{\tilde{u}}_j + \mathbf{C}\underline{\tilde{p}} = \underline{b}_B. \tag{4.3b}$$

4.2.2 Numerical fluxes

Below we give the numerical fluxes for the operators of the momentum equation, i.e. convection, diffusion and the pressure gradient, as well as for the divergence operator of the continuity equation.

4.2.2.1 Convective operator

For the convective operator we use the local Lax-Friedrichs flux (cf. (Hesthaven & Warburton 2007, Section 2.3)), which is given by

$$\begin{aligned} \hat{F}^C_{ij} &= \{u_{hi}u_{hj}\} + \frac{1}{2}\Lambda_{K,e} [\![u_{hi}]\!] \\ &= \begin{cases} \frac{1}{2}\left(u^+_{hi}u^+_{hj} + u^-_{hi}u^-_{hj}\right) + \frac{1}{2}\Lambda_{K,e}\left(u^+_{hi}n^+_j + u^-_{hi}n^-_j\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ \frac{1}{2}\left(u^+_{hi}u^+_{hj} + u_{Di}u_{Dj}\right) + \frac{1}{2}\Lambda_{K,e}\left(u^+_{hi} - u_{Di}\right)n^+_j, & \text{on } e \subseteq \partial\Omega_D, \\ u^+_{hi}u^+_{hj}, & \text{on } e \subseteq \partial\Omega_N \cup \Omega_{PO}. \end{cases} \end{aligned} \tag{4.4}$$

The parameter $\Lambda_{K,e}$ in (4.4) is calculated as

$$\Lambda_{K,e} = \max\left\{ |\lambda|; \lambda \in \operatorname{spec}\left(\mathbf{Q}\left(\overline{u^+_{hi}}\right)\right) \cup \operatorname{spec}\left(\mathbf{Q}\left(\overline{u^-_{hi}}\right)\right) \right\}, \tag{4.5}$$

where $\overline{u^\pm_{hi}}$ are the mean values of $u^\pm_{hi}$ in $K^\pm$ and

$$\mathbf{Q}(u_{hi}) = \frac{\partial u_{hi}u_{hk}n_k}{\partial u_{hj}} \tag{4.6}$$

is the flux Jacobian. Evaluating (4.5) leads to

$$\Lambda_{K,e} = \max\left\{2|\overline{u_{hk}^{+}} n_k^{+}|, 2|\overline{u_{hk}^{-}} n_k^{-}|\right\}. \tag{4.7}$$

4.2.2.2 Diffusive operator

The viscous terms (4.1c) are discretized applying the SIP method. A unified framework for the analysis of various Discontinuous Galerkin Methods for elliptic problems can be found in (Arnold, Brezzi, Cockburn & Marini 2002). The numerical flux for the SIP is given by

$$\begin{aligned}
\hat{F}_i^D &= \left\{\frac{\partial u_{hi}}{\partial x_j}\right\}[\![v^{l,m}]\!] + \left\{\frac{\partial v^{l,m}}{\partial x_j}\right\}[\![u_{hi}]\!] - \eta[\![u_{hi}]\!][\![v^{l,m}]\!] \\
&= \begin{cases}
\dfrac{1}{2}\left(\dfrac{\partial u_{hi}^{+}}{\partial x_j} + \dfrac{\partial u_{hi}^{-}}{\partial x_j}\right)\left(\left(v^{l,m}\right)^{+} n_j^{+} + \left(v^{l,m}\right)^{-} n_j^{-}\right) \\
+ \dfrac{1}{2}\left(\dfrac{\partial \left(v^{l,m}\right)^{+}}{\partial x_j} + \dfrac{\partial \left(v^{l,m}\right)^{-}}{\partial x_j}\right)\left(u_{hi}^{+} n_j^{+} + u_{hi}^{-} n_j^{-}\right) \\
- \eta\left(u_{hi}^{+} n_j^{+} + u_{hi}^{-} n_j^{-}\right)\left(\left(v^{l,m}\right)^{+} n_j^{+} + \left(v^{l,m}\right)^{-} n_j^{-}\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\
\dfrac{\partial u_{hi}^{+}}{\partial x_j} n_j^{+}\left(v^{l,m}\right)^{+} + \dfrac{\partial \left(v^{l,m}\right)^{+}}{\partial x_j} n_j^{+}\left(u_{hi}^{+} - u_{Di}\right) \\
- \eta\left(u_{hi}^{+} - u_{Di}\right)\left(v^{l,m}\right)^{+}, & \text{on } e \subseteq \partial\Omega_D, \\
0, & \text{on } e \subseteq \partial\Omega_N, \\
\dfrac{\partial u_{hi}^{+}}{\partial x_j} n_j^{+}\left(v^{l,m}\right)^{+}, & \text{on } e \subseteq \partial\Omega_{PO},
\end{cases}
\end{aligned} \tag{4.8}$$

The penalty parameter in (4.8) is determined by

$$\eta = \begin{cases} \max(c_{K^+}, c_{K^-}), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ c_{K^l}, & \text{on } e \subseteq \partial\Omega_D \cup \partial\Omega_N, \end{cases} \tag{4.9}$$

where

$$c_{K^l} = C(k)\frac{A(\partial K^l \setminus \partial\Omega)/2 + A(\partial K^l \cap \partial\Omega)}{V(K^l)}. \tag{4.10}$$

In (4.10), A and V are the area and volume of element K^l for three-dimensional problems or the perimeter and area for two-dimensional problems. The remaining constant

$C(k)$ in (4.10) depends on the polynomial order k and the mesh type. Sharp values for triangles and tetrahedrons are derived in (Shahbazi 2005)

$$C(k) = \frac{(k+1)(k+d)}{d}, \tag{4.11}$$

and for quadrilaterals and hexahedrons in (Hillewaert 2013)

$$C(k) = (p+1)^2. \tag{4.12}$$

4.2.2.3 Gradient operator

The pressure gradient (4.1d) is discretized by a central difference flux, which is evaluated as follows

$$\begin{aligned}\hat{F}^P &= \{p_h\} \\ &= \begin{cases} \frac{1}{2}\left(p_h^+ + p_h^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ p_h^+, & \text{on } e \subseteq \partial\Omega_D, \\ 0, & \text{on } e \subseteq \partial\Omega_N, \\ p_D, & \text{on } e \subseteq \partial\Omega_{PO}. \end{cases}\end{aligned} \tag{4.13}$$

4.2.2.4 Divergence operator

The numerical flux $\hat{F}^{\mathrm{Div}}$ for the discrete divergence operator $\mathbf{B}_j$ is chosen to ensure the compatibility relation between the gradient and divergence operators, i.e. $\mathbf{A}_{P_i}^{\mathrm{T}} = -\mathbf{B}_i$. Furthermore, we use the pressure stabilization developed in (Cockburn, Kanschat, Schötzau & Schwab 2002), (Cockburn, Kanschat & Schötzau 2004) and (Cockburn et al. 2009) to stabilize the equal-order method. Then, the numerical flux for the divergence operator reads [5]

$$\begin{aligned}\hat{F}_j^{\mathrm{Div}} &= \left\{u_{hj}\right\} + \kappa\, Re\, h_E \left[\!\left[p\right]\!\right] \\ &= \begin{cases} \frac{1}{2}\left(u_{hj}^+ + u_{hj}^-\right) + \kappa\, Re\, h_E (p^+ n_j^+ + p^- n_j^-), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ u_{Dj}, & \text{on } e \subseteq \partial\Omega_D, \\ u_{hj}^+, & \text{on } e \subseteq \partial\Omega_N \cup \Omega_{PO}. \end{cases}\end{aligned} \tag{4.14}$$

[5] In (Klein et al. 2015) there was a typo in the definition of the numerical flux for the divergence at an outflow boundary $e \subseteq \partial\Omega_N \cup \Omega_{PO}$, which has been corrected in here.

For the mixed-order method $\kappa = 0$, i.e. no pressure stabilization is needed and the matrix $\mathbf{C}$ in (4.1e) vanishes, whereas $\kappa = 1$ for the equal-order method. To ensure stability for the equal-order method the stabilization parameter $\lambda_E := Re\, h_E$ in (4.14) has to be strictly positive, where h_E is a local length scale. For $d = 2$ we take the length of the corresponding edge and for $d = 3$ the face diameter. The effect of different choices for the stabilization parameter λ_E on the accuracy of the method and the condition numbers of the matrices is discussed in (Cockburn et al. 2002) and (Cockburn et al. 2004).

4.3 Temporal discretization

The semidiscrete system of equations (4.3a)-(4.3b) still needs to be discretized in time. Therefore, we divide the time interval $[0, T]$ into uniform time steps Δt and introduce the following notation $\underline{\tilde{u}}_i^n := \underline{\tilde{u}}_i(n\Delta t)$, where $n = 0, 1, \ldots, T/\Delta t$. Then, applying the backward differentiation formulae (BDF) of order s yields

$$\frac{\beta_0}{\gamma \Delta t} \underline{\tilde{u}}_i^{n+1} + \frac{1}{\gamma \Delta t} \sum_{\alpha=1}^{s} \beta_\alpha \underline{\tilde{u}}_i^{n+1-\alpha} + \left(\mathbf{A}_C(\underline{\tilde{u}}^{n+1}) - \frac{1}{Re} \mathbf{A}_D \right) \underline{\tilde{u}}_i^{n+1} + \mathbf{A}_{P_i} \underline{\tilde{p}}^{n+1} = \underline{b}_i^{n+1}, \tag{4.15a}$$

$$\mathbf{B}_j \underline{\tilde{u}}_j^{n+1} + \mathbf{C} \underline{\tilde{p}}^{n+1} = \underline{b}_B, \tag{4.15b}$$

where the current and previous time steps are denoted by the superscripts $n+1$ and $n+1-\alpha$, respectively. The coefficients β_0, β_α and γ in (4.15a)-(4.15b) of the BDF schemes for order s from 1 to 4 are given in Table 4.1, cf. e.g. (Hairer & Wanner 1996). For ease of notation we summarize the second term of (4.15a) including all previous time steps in [6]

$$\underline{\sigma}_{u_i}^n := \sum_{\alpha=1}^{s} \beta_\alpha \underline{\tilde{u}}_i^{n+1-\alpha}. \tag{4.16}$$

After discretization in space and time we end up with the following non-linear system of equations without loss of generality for $d = 2$

$$\begin{pmatrix} \frac{\beta_0}{\gamma \Delta t} \mathbf{I} + \mathbf{A}_C^{n+1} - \frac{1}{Re} \mathbf{A}_D & 0 & \mathbf{A}_{P_1} \\ 0 & \frac{\beta_0}{\gamma \Delta t} \mathbf{I} + \mathbf{A}_C^{n+1} - \frac{1}{Re} \mathbf{A}_D & \mathbf{A}_{P_2} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^{n+1} \\ \underline{\tilde{u}}_2^{n+1} \\ \underline{\tilde{p}}^{n+1} \end{pmatrix} = \begin{pmatrix} \underline{b}_1^{n+1} - \frac{1}{\gamma \Delta t} \underline{\sigma}_{u_1}^n \\ \underline{b}_2^{n+1} - \frac{1}{\gamma \Delta t} \underline{\sigma}_{u_2}^n \\ \underline{b}_B \end{pmatrix}, \tag{4.17}$$

[6] Note that there was a typo in the definition of $\underline{\sigma}_i^n$ in (Klein et al. 2015), which has been corrected in here.

where $\mathbf{A}_C^{n+1} := \mathbf{A}_C(\tilde{\underline{u}}^{n+1})$. The SIMPLE algorithm for solving (4.17) is given in Section 4.5.

Table 4.1: Coefficients for the BDF schemes of different orders s.

order s	γ	β_0	β_1	β_2	β_3	β_4
1	1	1	-1			
2	2	3	-4	1		
3	6	11	-18	9	-2	
4	12	25	-48	36	-16	3

4.4 Boundary conditions

The boundary conditions for incompressible flows (2.4a)-(2.4d) will be satisfied in a weak sense by the choice of the numerical fluxes at the domain boundaries. To see which boundary conditions are implied by the numerical fluxes of Section 4.2.2 we start from the weak form of the steady incompressible momentum equations

$$\int_{\partial K^l} \hat{F}_{ij}^C n_{ej} v^{l,m} - \frac{1}{Re}\hat{F}_i^D + \hat{F}^P n_{ei} v^{l,m} ds \\ - \int_{K^l} u_{hi}u_{hj}\frac{\partial v^{l,m}}{\partial x_j} - \frac{1}{Re}\frac{\partial u_{hi}}{\partial x_j}\frac{\partial v^{l,m}}{\partial x_j} + p_h \frac{\partial v^{l,m}}{\partial x_i} d\mathbf{x} = 0. \tag{4.18}$$

For ease of notation in this section, we refer to the inner and outer cell values at some edge $e \in \partial K^l$ by the superscripts "+" and "-", respectively. Then, the following relation holds for the normal vectors $n_{ej} = n^+ = -n^-$. For the test functions $v^{l,m}$ we drop the superscript indicating inner and outer values, since the test functions have only support inside the associated cell K^l. Next, we do integration by parts for the volume integral of (4.18) once again to derive the strong form

$$\int_{\partial K^l} \left(\hat{F}_{ij}^C - u_{hi}^+ u_{hj}^+\right) n_{ej} v^{l,m} - \frac{1}{Re}\left(\hat{F}_i^D - \frac{\partial u_{hi}^+}{\partial x_j} n_{ej} v^{l,m}\right) + \left(\hat{F}^P - p_h^+\right) n_{ei} v^{l,m} ds \\ + \int_{K^l} \left(\frac{\partial u_{hi}u_{hj}}{\partial x_j} - \frac{1}{Re}\frac{\partial^2 u_{hi}}{\partial x_j \partial x_j} + \frac{\partial p_h}{\partial x_i}\right) v^{l,m} d\mathbf{x} = 0. \tag{4.19}$$

Note that for the edge integrals of this second integration by parts only inner values are taken. For the approximate solution u_{hi} we enforce that the volume integral of (4.19) shall vanish, cf. (3.4). Therefore, the surface integral of (4.19) has also to be zero

$$\int_{\partial K^l} \left(\hat{F}_{ij}^C - u_{hi}^+ u_{hj}^+\right) n_{ej} v^{l,m} - \frac{1}{Re}\left(\hat{F}_i^D - \frac{\partial u_{hi}^+}{\partial x_j} n_{ej} v^{l,m}\right) + \left(\hat{F}^P - p_h^+\right) n_{ei} v^{l,m} ds \stackrel{!}{=} 0, \tag{4.20}$$

which is the starting point for the derivation of the boundary conditions. By (4.20) we can also check the consistency of the numerical fluxes. To do that, we evaluate (4.20) at the inner edges by inserting the numerical fluxes (4.4), (4.8) and (4.13)

$$\begin{aligned}\int_{\Gamma_I} &\frac{1}{2}\left(u_{hi}^- u_{hj}^- - u_{hi}^+ u_{hj}^+\right) n_{ej} v^{l,m} + \frac{1}{2}\Lambda_{K,e}\left(u_{hi}^+ - u_{hi}^-\right) v^{l,m} \\ &- \frac{1}{Re}\Bigg(\frac{1}{2}\left(\frac{\partial u_{hi}^-}{\partial x_j} - \frac{\partial u_{hi}^+}{\partial x_j}\right) n_{ej} v^{l,m} + \frac{1}{2}\frac{\partial v^{l,m}}{\partial x_j}\left(u_{hi}^+ - u_{hi}^-\right) n_{ej} \\ &\qquad - \eta\left(u_{hi}^+ - u_{hi}^-\right) v^{l,m}\Bigg) + \frac{1}{2}\left(p_h^- - p_h^+\right) n_{ei} v^{l,m} ds = 0. \end{aligned} \tag{4.21}$$

We find that (4.21) is fulfilled, if jumps in u_{hi} and $\frac{\partial u_{hi}}{\partial x_j} n_{ej}$ as well as in p_h are zero. Hence, we can conclude that the numerical fluxes are consistent. Inserting the numerical fluxes (4.4), (4.8) and (4.13) at a Dirichlet boundary in (4.19) yields

$$\begin{aligned}\int_{\partial\Omega_D} &\frac{1}{2}\left(u_{Di} u_{Dj} - u_{hi}^+ u_{hj}^+\right) n_{ej} v^{l,m} + \frac{1}{2}\Lambda_{K,e}\left(u_{hi}^+ - u_{Di}\right) v^{l,m} \\ &- \frac{1}{Re}\Bigg(\frac{\partial v^{l,m}}{\partial x_j}\left(u_{hi}^+ - u_{Di}\right) n_{ej} - \eta\left(u_{hi}^+ - u_{Di}\right) v^{l,m}\Bigg) ds = 0, \end{aligned} \tag{4.22}$$

which implies the corresponding boundary condition.

$$u_{hi}^+ = u_{Di} \quad \text{on } e \subseteq \partial\Omega_D. \tag{4.23}$$

Applying the same procedure at $\partial\Omega_N$, we get

$$\int_{\partial\Omega_N} \frac{1}{Re}\frac{\partial u_{hi}^+}{\partial x_j} n_{ej} v^{l,m} - p_h^+ n_{ei} v^{l,m} ds = 0, \tag{4.24}$$

which implies the well-known outflow boundary condition

$$\frac{1}{Re}\frac{\partial u_{hi}^+}{\partial x_j} n_{ej} - p_h^+ n_{ei} = 0, \quad \text{on } e \subseteq \partial\Omega_N. \tag{4.25}$$

Even though this outflow boundary condition (4.25) is not completely physical, it performs well in many cases and is widely used in the Finite Element community, cf. (Gresho 1991), as well as for Discontinuous Galerkin Methods. The outflow boundary condition (4.25) introduces some coupling between the velocity gradient in the normal direction and the pressure at Ω_N. Choosing different fluxes at $\partial\Omega_{PO}$, cf. (4.4), (4.8) and (4.13), we can realize another boundary condition

$$\int_{\partial\Omega_{PO}} \left(p_D - p_h^+\right) n_{ei} v^{l,m} ds = 0, \tag{4.26}$$

respectively

$$p_h^+ = p_D \quad \text{on } e \subseteq \partial\Omega_{PO}, \tag{4.27}$$

which only includes the pressure with no constraint for the velocity.

4.5 SIMPLE algorithm

In this section, we discuss the SIMPLE algorithm for solving the discretized non-linear system of equations (4.17) for the unknowns at the new time step $\underline{\tilde{u}}_1(t_{n+1}) = \underline{\tilde{u}}_1^{n+1}$, $\underline{\tilde{u}}_2(t_{n+1}) = \underline{\tilde{u}}_2^{n+1}$ and $\underline{\tilde{p}}(t_{n+1}) = \underline{\tilde{p}}^{n+1}$. The SIMPLE algorithm was first proposed in the context of the Finite Difference Method, cf. (Patankar & Spalding 1972). Starting with the values from the previous time step, we introduce an iterative process:

$$\begin{aligned}\{\underline{\tilde{u}}(t_n), \underline{\tilde{p}}(t_n)\} =: \{\underline{\tilde{u}}_i^0, \underline{\tilde{p}}^0\} \mapsto \ldots \mapsto \{\underline{\tilde{u}}_i^\vartheta, \underline{\tilde{p}}^\vartheta\} \mapsto \\ \mapsto \{\underline{\tilde{u}}_i^{\vartheta+1}, \underline{\tilde{p}}^{\vartheta+1}\} \mapsto \ldots \xrightarrow{\lim} \{\underline{\tilde{u}}(t_{n+1}), \underline{\tilde{p}}(t_{n+1})\},\end{aligned}$$

where the superscript ϑ represents the inner SIMPLE iterations for each time step and for $\vartheta = 0$ the values from the previous time step t_n are taken. The convective part gets linearized by setting $\mathbf{A}_C^\vartheta := \mathbf{A}_C(\underline{\tilde{u}}^\vartheta)$. Then, we obtain the following linear system of equations for the unknowns at iteration step $\vartheta + 1$

$$\begin{pmatrix} \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^\vartheta - \frac{1}{Re}\mathbf{A}_D & 0 & \mathbf{A}_{P_1} \\ 0 & \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^\vartheta - \frac{1}{Re}\mathbf{A}_D & \mathbf{A}_{P_2} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^{\vartheta+1} \\ \underline{\tilde{u}}_2^{\vartheta+1} \\ \underline{\tilde{p}}^{\vartheta+1} \end{pmatrix} = \begin{pmatrix} \underline{b}_1^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n \\ \underline{b}_2^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n \\ \underline{b}_B \end{pmatrix}. \tag{4.28}$$

Next, we use a decomposition of the unknowns into intermediate, $\underline{\tilde{u}}_i^*$ and $\underline{\tilde{p}}^* := \underline{\tilde{p}}^\vartheta$, and correction components, $\underline{\tilde{u}}_i'$ and $\underline{\tilde{p}}'$,

$$\underline{\tilde{u}}_i^{\vartheta+1} = \underline{\tilde{u}}_i^* + \underline{\tilde{u}}_i', \tag{4.29}$$

$$\underline{\tilde{p}}^{\vartheta+1} = \underline{\tilde{p}}^* + \underline{\tilde{p}}' := \underline{\tilde{p}}^\vartheta + \underline{\tilde{p}}'. \tag{4.30}$$

Now, in the first step of the SIMPLE algorithm we sequentially solve the linear and decoupled system of equations for the intermediate velocity components $\underline{\tilde{u}}_1^*$ and $\underline{\tilde{u}}_2^*$

$$\begin{pmatrix} \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^\vartheta - \frac{1}{Re}\mathbf{A}_D & 0 \\ 0 & \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^\vartheta - \frac{1}{Re}\mathbf{A}_D \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^* \\ \underline{\tilde{u}}_2^* \end{pmatrix} = \begin{pmatrix} \underline{b}_1^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n \\ \underline{b}_2^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n \end{pmatrix} - \begin{pmatrix} \mathbf{A}_{P_1} \\ \mathbf{A}_{P_2} \end{pmatrix} \underline{\tilde{p}}^\vartheta. \tag{4.31}$$

Inserting the decomposition for velocity (4.29) and pressure (4.30) into (4.28) and subtracting the predictor equation (4.31) yields a linear system of equations for the correction components

$$\begin{pmatrix} \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^{\vartheta} - \frac{1}{Re}\mathbf{A}_D & 0 & \mathbf{A}_{P_1} \\ 0 & \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^{\vartheta} - \frac{1}{Re}\mathbf{A}_D & \mathbf{A}_{P_2} \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{C} \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1' \\ \underline{\tilde{u}}_2' \\ \underline{\tilde{p}}' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \underline{b}_B - \mathbf{B}_1\underline{\tilde{u}}_1^* - \mathbf{B}_2\underline{\tilde{u}}_2^* - \mathbf{C}\underline{\tilde{p}}^{\vartheta} \end{pmatrix}, \tag{4.32}$$

where we introduce the following approximation

$$\frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \underbrace{\mathbf{A}_C^{\vartheta} - \frac{1}{Re}\mathbf{A}_D}_{\approx \mathbf{I}} \approx \frac{\beta_0 + \gamma\Delta t}{\gamma\Delta t}\mathbf{I}. \tag{4.33}$$

Note that the choice of this approximation does not affect the solution, it only affects the convergence behavior of the algorithm. Taking a better approximation like the diagonal or block diagonal matrix of $\mathbf{A}_C^{\vartheta} - \frac{1}{Re}\mathbf{A}_D$ can have a positive effect on the number of SIMPLE iterations needed to reach the specified convergence criterion, cf. the discussion for variable density flows in Section 6.5. Using (4.33), we can rewrite the momentum equations of (4.32) for the correction components of the velocity

$$\begin{pmatrix} \underline{\tilde{u}}_1' \\ \underline{\tilde{u}}_2' \end{pmatrix} = -\frac{\gamma\Delta t}{\beta_0 + \gamma\Delta t} \begin{pmatrix} \mathbf{A}_{P_1} \\ \mathbf{A}_{P_2} \end{pmatrix} \underline{\tilde{p}}'. \tag{4.34}$$

By inserting (4.34) into the continuity equation of (4.32), we obtain an equation for the pressure correction

$$\left(\mathbf{B}_1\mathbf{A}_{P_1} + \mathbf{B}_2\mathbf{A}_{P_2} - \frac{\beta_0 + \gamma\Delta t}{\gamma\Delta t}\mathbf{C} \right) \underline{\tilde{p}}' = \frac{\beta_0 + \gamma\Delta t}{\gamma\Delta t} \left(-\underline{b}_B + \mathbf{B}_1\underline{\tilde{u}}_1^* + \mathbf{B}_2\underline{\tilde{u}}_2^* + \mathbf{C}\underline{\tilde{p}}^{\vartheta} \right). \tag{4.35}$$

Having solved (4.35) for $\underline{\tilde{p}}'$, we can update the velocity by (4.29) using (4.34) and the pressure by (4.30).

Next, in Section 4.5.1 we comment on the reference point for the solution of the pressure, in Section 4.5.2 we will discuss the symmetric interior penalty method for the pressure correction, in Section 4.5.3 we will comment on the mass conservation of the different methods, in Section 4.5.4 we will give the formulae for under-relaxation and in Section 4.5.5 we will summarize the algorithm.

4.5.1 Reference point pressure

For incompressible flows the solution for the pressure is only determined up to a constant $C \in \mathbb{R}$. Numerically the pressure level can either be determined by a boundary condition or by a reference point. In case of the two outflow boundary conditions (2.4b) and (2.4c) the pressure level is set by these conditions and no reference point is needed. For test cases with only Dirichlet boundary conditions for the velocity (2.4a)

and/or periodic boundaries (2.4d) the pressure level needs to be fixed. For that, the zeroth order coefficient in some arbitrary cell is set to zero by manipulating (4.35). For $h-$ and $p-$convergence studies it is important to assure that the global mean values of the numerical solution p_h and the exact solution p are the same. This is done in a post-processing step by shifting the calculated pressure by a constant such that $\int_\Omega p_h \, d\mathbf{x} = \int_\Omega p \, d\mathbf{x}$.

4.5.2 Symmetric interior penalty method for pressure correction

Recognizing that $\mathbf{P} := \mathbf{B}_1\mathbf{A}_{P_1} + \mathbf{B}_2\mathbf{A}_{P_2}$ in the equation for the pressure correction (4.35) represents a discrete version of the Laplace operator, we obtain an alternative equation for the pressure correction by replacing $\mathbf{P}$ with an equivalent matrix stemming from the symmetric interior penalty method (cf. e.g. (Arnold et al. 2002)) for the Laplace operator. The boundary conditions for the Laplacian using the SIP method are homogeneous Neumann and Dirichlet at $\partial\Omega_D$ and $\partial\Omega_N$, respectively. Note that Dirichlet and Neumann boundary conditions are switched for the Laplacian compared to the Navier-Stokes equations. The idea of replacing $\mathbf{P}$ in the equation for the pressure correction with the SIP discretization originates from (Shahbazi et al. 2007), where a similar replacement is used for the pressure operator in the context of an algebraic splitting scheme. The main motivation for using this replacement is to reduce the stencil size. While the matrix $\mathbf{P}$ has got a stencil involving the first and second layer of neighbors, the SIP method yields a reduced stencil involving only the immediate neighbors. The penalty parameter for the SIP method is given by (4.9)-(4.12). In the following, we refer to the approach applying the original equation for the pressure correction (4.35) as the SIMPLE-LDG method due to its similarity to the Local Discontinuous Galerkin Method (Cockburn & Shu 1998) for the Laplace operator. To the alternative formulation, using the symmetric interior penalty method for the pressure correction, we refer to as the SIMPLE-SIP method. In Chapter 5 we will compare both methods with respect to accuracy, numerical stability and efficiency in terms of the number of SIMPLE iterations needed to reach the specified convergence criterion. The comparison will be done for the mixed-order as well as the equal-order formulation.

4.5.3 Note on mass conservation

Originally, the SIMPLE algorithm is constructed to yield a velocity field, which is divergence-free up to machine accuracy after each single SIMPLE iteration. This is true for the mixed-order SIMPLE-LDG method and can readily be shown by

$$\begin{aligned}
\mathbf{B}_1\underline{\tilde{u}}_1^{\vartheta+1} + \mathbf{B}_2\underline{\tilde{u}}_2^{\vartheta+1} - \underline{b}_B &= \mathbf{B}_1(\underline{\tilde{u}}_1^{*} + \underline{\tilde{u}}_1') + \mathbf{B}_2(\underline{\tilde{u}}_2^{*} + \underline{\tilde{u}}_2') - \underline{b}_B \\
&= \mathbf{B}_1\underline{\tilde{u}}_1^{*} + \mathbf{B}_2\underline{\tilde{u}}_2^{*} - \underline{b}_B + \mathbf{B}_1\underline{\tilde{u}}_1' + \mathbf{B}_2\underline{\tilde{u}}_2' \\
&\overset{(4.34)}{=} \mathbf{B}_1\underline{\tilde{u}}_1^{*} + \mathbf{B}_2\underline{\tilde{u}}_2^{*} - \underline{b}_B - \frac{\gamma\Delta t}{\beta_0 + \gamma\Delta t}\left(\mathbf{B}_1\mathbf{A}_{P_1} + \mathbf{B}_2\mathbf{A}_{P_2}\right)\underline{\tilde{p}}' \\
&\overset{(4.35)}{=} 0. \qquad (4.36)
\end{aligned}$$

Note the last line in (4.36) is only valid for the mixed-order SIMPLE-LDG method, since the matrix $\mathbf{C}$ in (4.35) vanishes in that case. For the mixed-order SIMPLE-SIP method the pressure stabilization matrix vanishes as well, but the operator $\mathbf{B}_1\mathbf{A}_{P_1} + \mathbf{B}_2\mathbf{A}_{P_2}$ in the equation for the pressure correction (4.35) is replaced by the SIP discretization. In turn (4.36) is not valid for that case. For the equal-order SIMPLE-LDG and the equal-order SIMPLE-SIP methods the presence of the pressure stabilization matrix leads to a velocity field, which is not divergence-free like already pointed out in (Cockburn et al. 2009).

4.5.4 Under-relaxation

For most test cases, under-relaxation is required in order to get a convergent scheme. Therefore, the equation for the intermediate velocity components (4.31) is replaced by the implicit relaxation

$$\left[\left(\frac{1-\alpha_u}{\alpha_u}\right)\frac{\beta_0+\gamma\Delta t}{\gamma\Delta t}\mathbf{I} + \frac{\beta_0}{\gamma\Delta t}\mathbf{I} + \mathbf{A}_C^{\vartheta} - \frac{1}{Re}\mathbf{A}_D\right]\underline{\tilde{u}}_i^* = \underline{b}_i^{\vartheta} - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_i}^n - \mathbf{A}_{P_i}\underline{\tilde{p}}^{\vartheta} + \left(\frac{1-\alpha_u}{\alpha_u}\right)\frac{\beta_0+\gamma\Delta t}{\gamma\Delta t}\underline{\tilde{u}}_i^{\vartheta}, \quad (4.37)$$

where the relaxation factor for the velocity α_u can be varied between between 0 and 1. Setting $\alpha_u = 1$ in (4.37), one gets Eq. (4.31), i.e. no relaxation, and in the limit $\alpha_u \to 0$ we obtain $\underline{\tilde{u}}_i^* = \underline{\tilde{u}}_i^{\vartheta}$, i.e. no change in the solution. The relaxation for the pressure is done explicitly instead

$$\underline{\tilde{p}}^{\vartheta+1} = \underline{\tilde{p}}^{\vartheta} + \alpha_p\underline{\tilde{p}}', \quad (4.38)$$

with the relaxation factor $\alpha_p \in (0, 1]$ for the pressure.

4.5.5 Summary of the SIMPLE algorithm

Solve (4.17) for each time step by the iterative process:

$$\{\underline{\tilde{u}}(t_n), \underline{\tilde{p}}(t_n)\} =: \{\underline{\tilde{u}}_i^0, \underline{\tilde{p}}^0\} \mapsto \ldots \mapsto \{\underline{\tilde{u}}_i^{\vartheta}, \underline{\tilde{p}}^{\vartheta}\} \mapsto \mapsto \{\underline{\tilde{u}}_i^{\vartheta+1}, \underline{\tilde{p}}^{\vartheta+1}\} \mapsto \ldots \xrightarrow{\lim} \{\underline{\tilde{u}}(t_{n+1}), \underline{\tilde{p}}(t_{n+1})\},$$

until the convergence criterion $\|p_h'\|_{L^2(\Omega)} \leq \epsilon_p \ \wedge \ \|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} \leq \epsilon_u$, where ϵ_p and ϵ_u are predefined tolerances, is fulfilled. The steps for one SIMPLE iteration within one time step are:

1. Solve (4.37) sequentially for the intermediate velocity components $\underline{\tilde{u}}_i^*$.
2. Calculate the pressure correction $\underline{\tilde{p}}'$ using either the SIMPLE-LDG method, i.e. solve (4.35), or the SIMPLE-SIP method, i.e. replace the operator $\mathbf{B}_1\mathbf{A}_{P_1} + \mathbf{B}_2\mathbf{A}_{P_2}$

in (4.35) with the SIP discretization. Note that the pressure stabilization matrix $\mathbf{C}$ in (4.35) vanishes for the mixed-order formulation.

3. Update the pressure via (4.38) and the velocity via (4.29) using (4.34).

Note that for steady incompressible flows the algorithm is exactly the same except for the outer time loop.

5 Numerical results for incompressible flows

In this chapter, we present numerical results for steady and unsteady incompressible flows. In the previous chapter we have discussed the spatial and temporal discretization as well as the SIMPLE algorithm for solving the non-linear discrete system of equations. Our aim here is to verify the correct implementation in BoSSS. Besides, we will also assess the performance and stability of the SIMPLE algorithm. The results for steady incompressible flows in Section 5.1 have been previously published in (Klein et al. 2013, Sect. 4) and the results for unsteady incompressible flows in Section 5.2 have been published before in (Klein et al. 2015, Sect. 4). In what follows, only minor formulations have been changed. We shall also acknowledge the work of Keil (2012), who contributed to the results of the Taylor vortex flow in Section 5.2.1 and of the two-dimensional flow past a square cylinder in Section 5.2.3.1. Finally, the simulations to determine the critical Reynolds number for the two- and three-dimensional flow past a square cylinder presented in Section 5.2.3.2 have been performed by Fischer (2014).

5.1 Steady incompressible flows

Below, three test cases are presented for steady incompressible flows: the Kovasznay flow in Section 5.1.2, the flow into a corner in Section 5.1.3 and the backward-facing step flow in Section 5.1.4. Due to rather small problem sizes the sparse direct solver PARDISO, cf. (Schenk & Gärtner 2004) and (Schenk & Gärtner 2006), is used for all three test cases to solve the linear systems of equations within the SIMPLE algorithm. Unless otherwise stated, the initial guess for the SIMPLE iterations at $\vartheta = 0$ is set to zero, i.e. $\underline{\tilde{u}}_i^0 = 0$ and $\tilde{p}^0 = 0$. The identity matrix is applied for the approximation in the corrector step, cf. Eq. (4.33), and the mixed-order formulation is used for the spatial discretization. Before showing the results and the convergence rates of the DGM for the individual problems, the convergence behavior of the SIMPLE algorithm as a function of the relaxation factors α_u and α_p is discussed.

5.1.1 Convergence behavior and performance of the steady SIMPLE algorithm

The convergence behavior of the SIMPLE algorithm is examined for taking the identity matrix, cf. Eq. (4.33). The relaxation factors α_u and α_p are varied independently from 0.1 to 1.0 by steps of 0.1. The calculations are terminated when the L^2-norms of the pressure correction $\|p'_h\|_{L^2(\Omega)}$ and the change in each velocity component during one

SIMPLE iteration $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)}$ reach the specified tolerances, after a maximum number of iterations or in case of divergence (denoted as NaN). This convergence study is carried out for all test cases using the mixed-order formulation with degrees $k = 2$ for the velocity and $k' = 1$ for the pressure. The individual grid sizes, the tolerances for the pressure correction and for the change in the velocity as well as the maximum number of iterations are given in the tables below. The initial guess for velocity and pressure is set to zero for all problems.

From the results in the Tables 5.1-5.3 it can be seen that the fastest convergence is achieved for $\alpha_p = 1.0$ for all three test cases, which means that no under-relaxation for the pressure is needed for these test cases. This is most likely due to the choice of the identity matrix in (4.33), which results in rather small values for the pressure correction compared to other options like taking a diagonal or block diagonal matrix, and hence no under-relaxation is needed for the pressure. In Chapter 7 we will show results for variable density flows taking the diagonal and block diagonal option resulting in different optimal values for the relaxation factors similar to the FVM, where one usually applies the rule of thumb $\alpha_p = 1 - \alpha_u$ (Ferziger & Perić 2002) with typical values for α_u between 0.6 and 0.8 and for α_p between 0.2 and 0.4. In our tests taking the identity matrix, the optimal values for the relaxation factor of the velocity α_u differ for the different problems. For the two cases, where Dirichlet boundary conditions for the velocity are used on all boundaries, namely the Kovasznay flow and the flow into a corner, the optimal value for the relaxation factor of the velocity is $\alpha_u = 1.0$, which means that no under-relaxation at all is needed for these cases. Table 5.4 gives a summary of the optimal values for the relaxation factors together with the computational time measured on a Intel Core i5 Quad Core (3.10 GHz) with 8.00 GB main memory. Simulations are run on a single processor except for the direct solver PARDISO, which is OpenMP parallel.

Table 5.1: Number of SIMPLE iterations as a function of the relaxation factors α_u and α_p for the Kovasznay flow using a grid with 8×8 cells and tolerances for $\|p_h'\|_{L^2(\Omega)}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)}$ of 10^{-10}. The optimal value is bold-faced.

	α_p=0.1	α_p=0.2	α_p=0.3	α_p=0.4	α_p=0.5	α_p=0.6	α_p=0.7	α_p=0.8	α_p=0.9	α_p=1.0
α_u=0.1	>2500	>2500	>2500	>2500	>2500	>2500	2207	1931	1717	1545
α_u=0.2	>2500	>2500	>2500	>2500	>2500	2456	2105	1841	1637	1473
α_u=0.3	>2500	>2500	>2500	>2500	>2500	2417	2071	1811	1610	1448
α_u=0.4	>2500	>2500	>2500	>2500	>2500	2397	2054	1797	1596	1436
α_u=0.5	>2500	>2500	>2500	>2500	>2500	2385	2044	1788	1588	1429
α_u=0.6	>2500	>2500	>2500	>2500	>2500	2377	2037	1782	1583	1424
α_u=0.7	>2500	>2500	>2500	>2500	>2500	2372	2032	1777	1579	1421
α_u=0.8	>2500	>2500	>2500	>2500	>2500	2367	2028	1774	1576	1418
α_u=0.9	>2500	>2500	>2500	>2500	>2500	2364	2025	1771	1573	1415
α_u=1.0	>2500	>2500	>2500	>2500	>2500	2360	2022	1768	1571	**1413**

Table 5.2: Number of SIMPLE iterations as a function of the relaxation factors α_u and α_p for the flow into a corner using a grid with 6×6 cells and tolerances for $\|p'_h\|_{L^2(\Omega)}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)}$ of 10^{-5}. The optimal value is bold-faced.

	α_p=0.1	α_p=0.2	α_p=0.3	α_p=0.4	α_p=0.5	α_p=0.6	α_p=0.7	α_p=0.8	α_p=0.9	α_p=1.0
α_u=0.1	>2500	>2500	>2500	1947	1558	1299	1114	975	867	780
α_u=0.2	>2500	>2500	>2500	1937	1550	1292	1108	969	862	776
α_u=0.3	>2500	>2500	>2500	1934	1547	1290	1106	968	860	774
α_u=0.4	>2500	>2500	>2500	1932	1546	1289	1105	967	859	774
α_u=0.5	>2500	>2500	>2500	1931	1545	1288	1104	966	859	773
α_u=0.6	>2500	>2500	>2500	1931	1545	1287	1104	966	859	773
α_u=0.7	>2500	>2500	>2500	1930	1544	1287	1103	966	858	773
α_u=0.8	>2500	>2500	>2500	1930	1544	1287	1103	965	858	773
α_u=0.9	>2500	>2500	>2500	1929	1544	1287	1103	965	858	**772**
α_u=1.0	>2500	>2500	>2500	1929	1544	1287	1103	965	858	**772**

Table 5.3: Number of SIMPLE iterations as a function of the relaxation factors α_u and α_p for the backward-facing step flow using a grid with 832 cells and tolerances for $\|p'_h\|_{L^2(\Omega)}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)}$ of 10^{-6}. The optimal value is bold-faced.

	α_p=0.1	α_p=0.2	α_p=0.3	α_p=0.4	α_p=0.5	α_p=0.6	α_p=0.7	α_p=0.8	α_p=0.9	α_p=1.0
α_u=0.1	>1000	910	892	886	881	877	875	873	872	871
α_u=0.2	>1000	>1000	723	557	465	449	449	446	442	441
α_u=0.3	>1000	>1000	842	644	524	445	391	356	333	**317**
α_u=0.4	>1000	>1000	907	740	616	530	470	429	396	368
α_u=0.5	>1000	>1000	>1000	809	688	601	535	484	446	417
α_u=0.6	>1000	>1000	>1000	897	753	658	594	545	504	469
α_u=0.7	>1000	>1000	>1000	965	826	730	654	595	549	512
α_u=0.8	>1000	>1000	>1000	>1000	895	784	702	NaN	NaN	NaN
α_u=0.9	>1000	>1000	>1000	>1000	NaN	NaN	NaN	NaN	NaN	NaN
α_u=1.0	>1000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Table 5.4: Performance of the SIMPLE algorithm for the different test cases: Optimal values for the relaxation factors α_u and α_p, degrees of freedom (DOF) for velocity and pressure, number of SIMPLE iterations and computational time.

	α_u	α_p	DOF(u)	DOF(p)	# iterations	time [sec]
Kovasznay flow	1.0	1.0	384	192	1413	38.1
Flow into a corner	1.0	1.0	216	108	772	18.9
Backward-facing step flow	0.3	1.0	4992	2496	317	62.3

5.1.2 Kovasznay flow

The Kovasznay flow, cf. (Kovasznay 1948), which may be considered as the flow behind a two-dimensional grid, has the analytical solution

$$u_1 = 1 - \exp(\lambda x_1)\cos(2\pi x_2), \tag{5.1}$$

$$u_2 = \frac{\lambda}{2\pi}\exp(\lambda x_1)\sin(2\pi x_2), \tag{5.2}$$

$$p = -\frac{1}{2}\exp(2\lambda x_1) + C, \tag{5.3}$$

where $\lambda = \frac{Re}{2} - \sqrt{\frac{Re^2}{4} + 4\pi^2}$ and $C \in \mathbb{R}$ is an arbitrary constant. We consider the computational domain $\Omega = [-0.5, 1.5] \times [-0.5, 1.5]$ discretized by successively refined uniform Cartesian grids with $\Delta x_i = \Delta x_0 / 2^i$, where $i \in \mathbb{N}$ and $\Delta x_0 = 0.5$ belongs to the coarsest grid consisting of 4×4 cells. The analytical solution for the velocity is taken as Dirichlet boundary condition on all domain boundaries and a reference point for the pressure is set at $(x_1, x_2) = (1.5, -0.5)$. For the Reynolds number $Re = 20$ is chosen.

Simulations are carried out on four different grids. The polynomial orders are $k = 1, \ldots, 5$ for the velocity and $k' = 0, \ldots, 4$ for the pressure. The errors of the calculated velocity and pressure are measured with respect to the analytical solution (5.1)-(5.3) in the L^2-norm. In Fig. 5.1 the h-convergence of the velocity and the pressure is displayed. We observe the convergence rates $k + 1$ for the velocity and $k' + 1$ for the pressure, as expected.

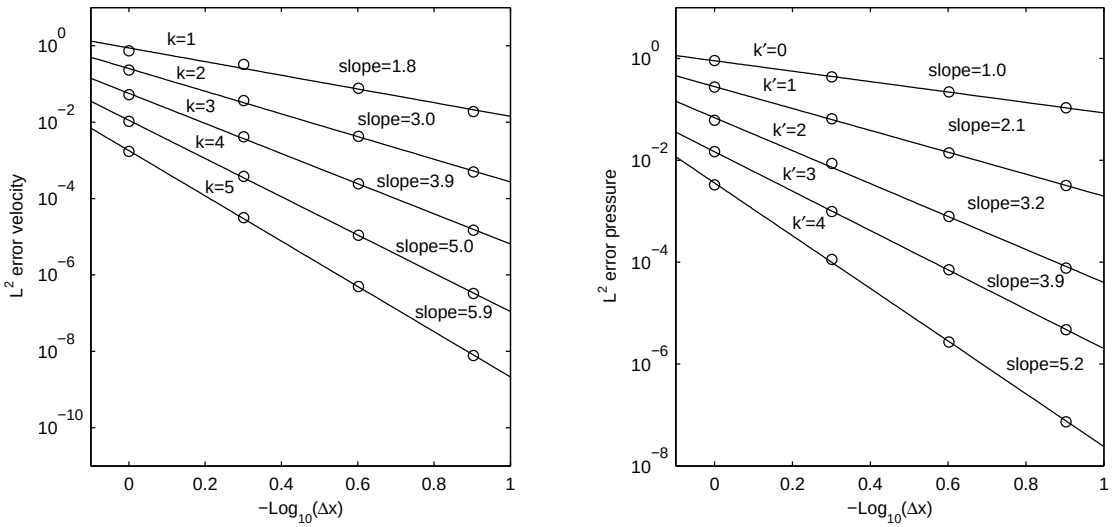


Figure 5.1: h-convergence for velocity (left) and pressure (right) for the Kovasznay flow. The best linear fits in a least square sense and the corresponding slopes of the L^2 errors are also plotted.

5.1.3 Flow into a corner

An analytical solution of the Navier-Stokes equations (2.1a)-(2.1b) is the so-called flow into a corner (Drazin & Riley 2006)

$$u_1 = \exp(-x_2 Re) - 1, \tag{5.4}$$
$$u_2 = \exp(-x_1 Re) - 1, \tag{5.5}$$
$$p = -\exp(-Re(x_1 + x_2)) + C. \tag{5.6}$$

Depending on the Reynolds number Re the solution is nearly singular (in the limit $Re \to \infty$) for u_1 as x_2 tends to zero, for u_2 as x_1 tends to zero and for p as x_1 and x_2 tend to zero. The computational domain used is $\Omega = [0, 0.05] \times [0, 0.05]$. To account for the nearly singular behavior the numerical mesh is refined towards $x_1 = 0$ and $x_2 = 0$. The coarsest grid consists of 6×6 cells. A refined grid with 24×24 cells is shown in Fig. 5.2 (left). Using the analytical solution (5.4)-(5.5), Dirichlet boundary conditions are taken for the velocity on all domain boundaries and the reference point for the pressure is set to zero at $(x_1, x_2) = (0.05, 0.05)$. For the Reynolds number we take $Re = 100$.

For this test case the convergence of the SIMPLE algorithm is relatively slow even when taking the optimal values for the relaxation factors $\alpha_u = 1.0$ and $\alpha_p = 1.0$ (cf. Table 5.2). This is most likely due to the nearly singular behavior and hence the steep gradients in velocity and pressure. In order to save iterations, the DG projection of the analytical solution is used as initial guess for the SIMPLE algorithm.

Simulations are performed on four different grids for the polynomial orders $k = 1, \ldots, 5$ and $k' = 0, \ldots, 4$, respectively. A plot of the streamlines and a contour plot of the pressure on a grid with 24×24 cells and for the polynomial orders $k = 5$ and $k' = 4$ are shown in Fig. 5.2. Again, a study of the h-convergence of the velocity and the pressure is carried out, which is shown in Fig. 5.3. As for the previous test cases the convergence rates are found to be $k + 1$ for the velocity and $k' + 1$ for the pressure.

5.1.4 Backward-facing step flow

The backward-facing step flow is widely used as a benchmark test for numerical simulations. We set up the calculation according to the experiment by Armaly, Durst, Pereira & Schönung (1983). The dimensionless geometry and a part of the non-uniform Cartesian grid close to the step is depicted in Fig. 5.4. The length of the channel is chosen to make sure that the flow is fully developed at the outlet, i.e. $\partial u_1 / \partial x_1 = 0$ and $\partial u_2 / \partial x_1 = 0$. The reference values are $L = 2h$ for the length scale, where h is the height of the channel at the inlet, and $U = 2/3 U_{\max}$ for the velocity, where $U_{\max}$ is two-thirds of the maximum of the streamwise velocity u_1 also taken at the inlet. Then the Reynolds number is $Re = 4 U_{\max} h / (3\nu)$, where ν is the kinematic viscosity. At the

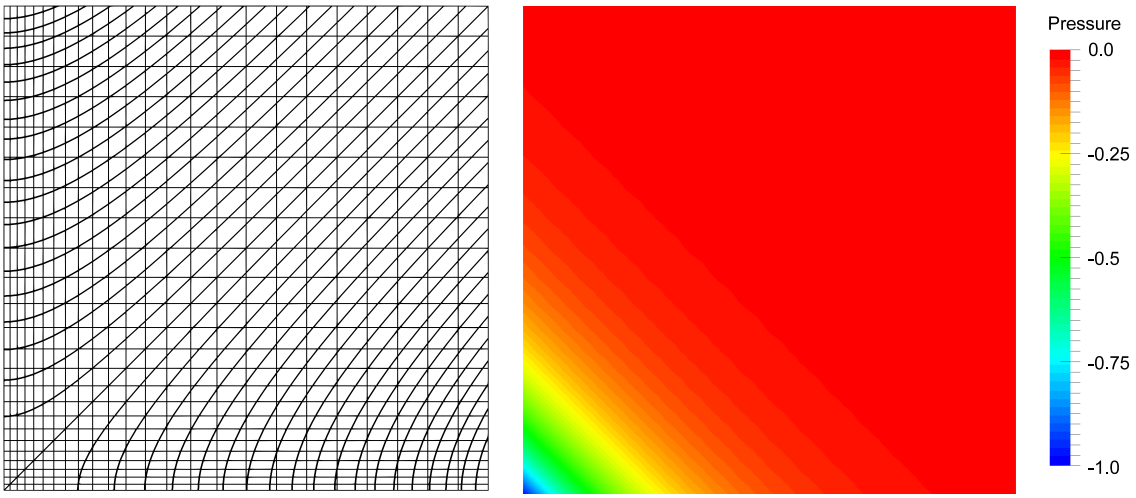


Figure 5.2: Mesh and streamlines (left) and contour plot of the pressure (right) on a grid with 24×24 cells with polynomial orders $k = 5$, $k' = 4$ for the flow into a corner test case.

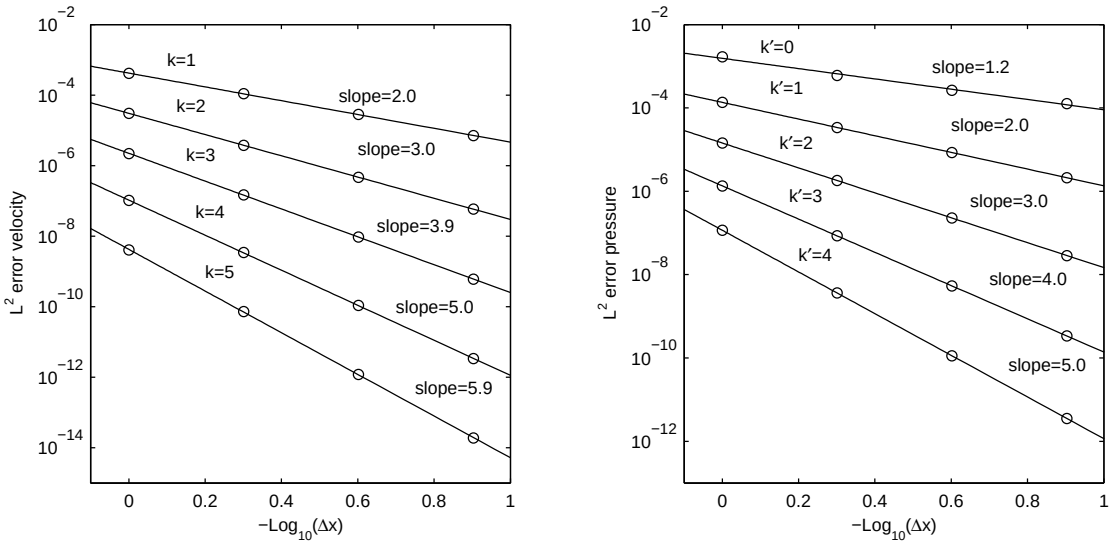


Figure 5.3: h-convergence for velocity (left) and pressure (right) for the flow into a corner test case. The best linear fits in a least square sense and the corresponding slopes of the L^2 errors are also plotted.

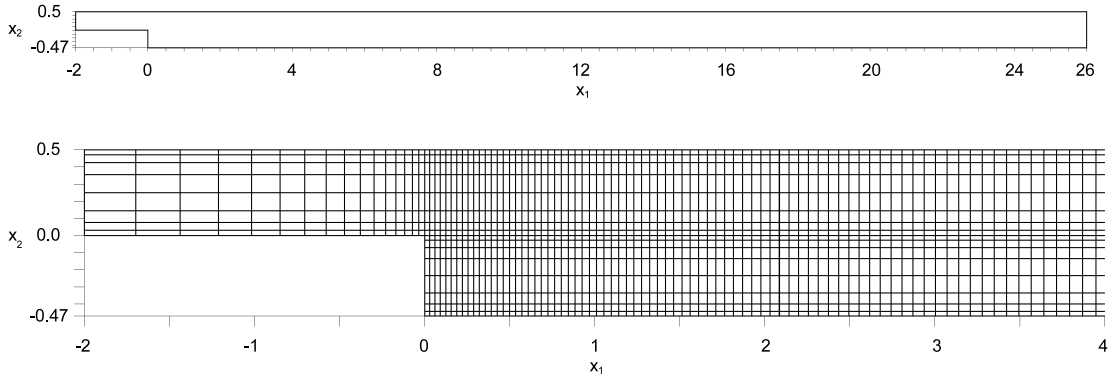


Figure 5.4: Geometry (top) and close-up of the grid (bottom) for the backward-facing step flow.

walls the no-slip condition is applied, the boundary conditions at the inlet ($x_1 = -2$) are

$$u_1(-2, x_2) = 12x_2(1 - 2x_2), \tag{5.7}$$
$$u_2(-2, x_2) = 0, \tag{5.8}$$

and at the outlet ($x_1 = 26$) the outflow boundary condition (2.4c) is used. The entire grid consists of 3328 cells and the polynomial orders are $k = 5$ for the velocity and $k' = 4$ for the pressure. For the Reynolds number we take $Re = 300$. Following (Armaly et al. 1983), for this Reynolds number the flow is still laminar, two-dimensional and there is only one recirculation zone. In Fig. 5.5 profiles of the velocity u_1 are plotted at different positions in the streamwise direction. Like in the experiment, we also find only one zone of recirculation in our calculation. Our point of reattachment is at $(x_R, y_R) = (3.1, -0.47)$, which agrees very well with the results reported in (Armaly et al. 1983). Fig. 5.6 and Fig. 5.7 show the contours of the pressure and the streamlines in the region close to the step, respectively. In the plot of the pressure the singularity at the corner of the step can be clearly observed.

5.2 Unsteady incompressible flows

In this section, the numerical results for three unsteady incompressible test cases are presented: the Taylor vortex flow, the Orr-Sommerfeld stability problem for plane Poiseuille flow and the flow past a square cylinder. Again, the sparse direct solver PARDISO (Schenk & Gärtner 2004, Schenk & Gärtner 2006) is used for the linear systems in the SIMPLE algorithm, except for the two- and three-dimensional simulations in Section 5.2.3.2, where iterative solvers from the Hypre library (Falgout, Jones & Yang 2006) are taken. The momentum equations (4.37) are solved by the generalized minimal residual (GMRES) method and the pressure correction equation (4.35) by the conjugate gradient (CG) method.

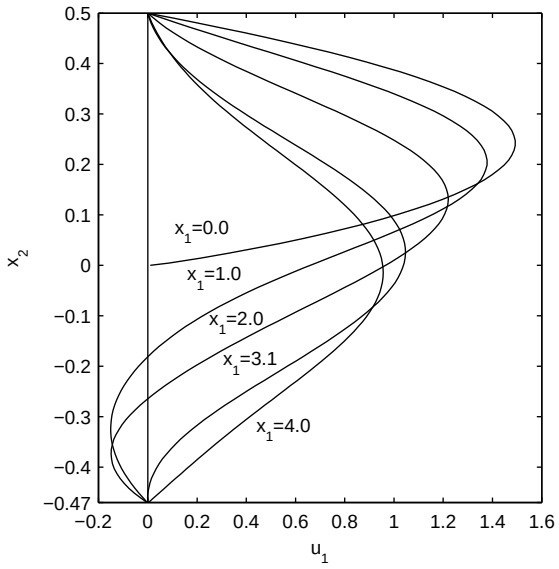


Figure 5.5: Profiles of the streamwise velocity at different locations for the backward-facing step flow.

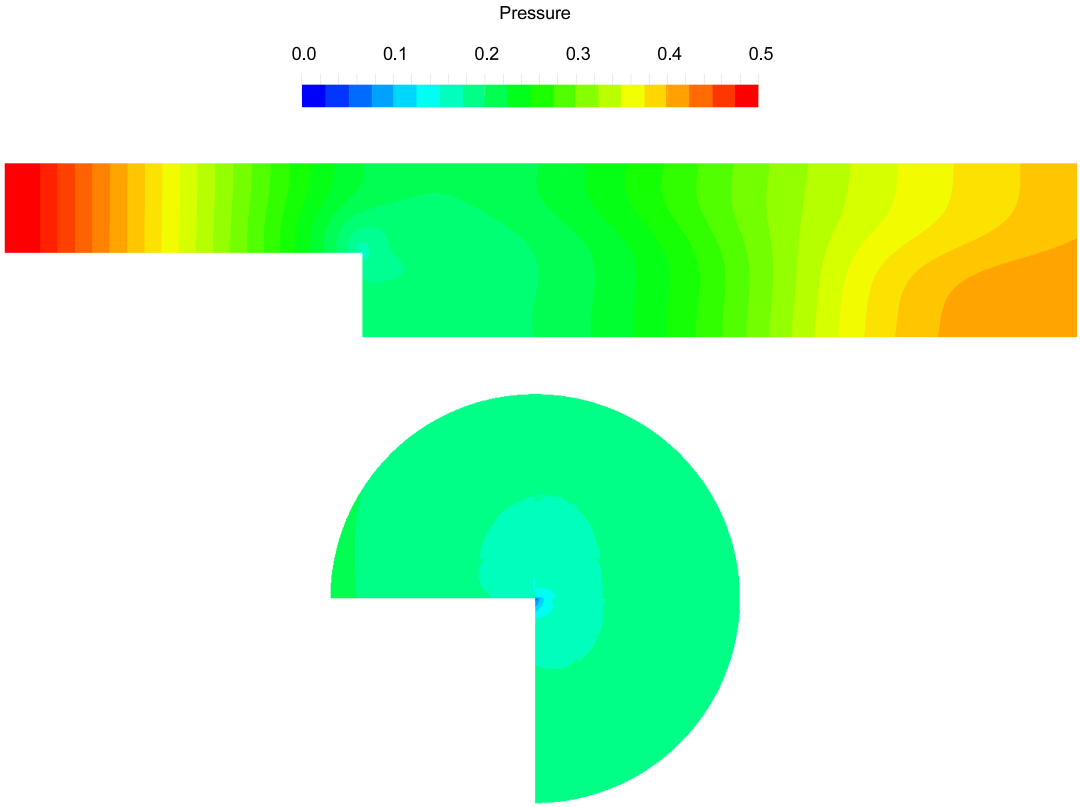


Figure 5.6: Contour plot of the pressure close to the step for the backward-facing step flow (top) and zoom at the corner of the step showing the singularity of the solution for the pressure (bottom).

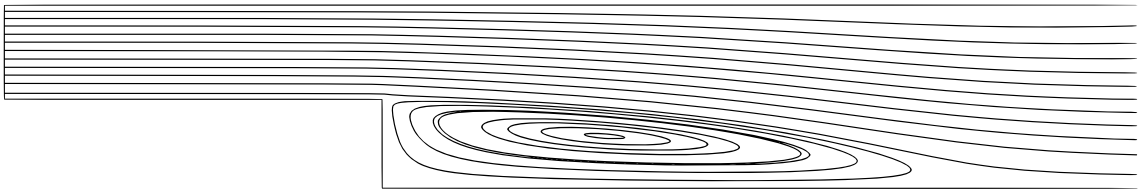


Figure 5.7: Streamlines close to the step for the backward-facing step flow.

For the Taylor vortex in Section 5.2.1 flow we examine the temporal and spatial accuracy for the mixed- and equal-order formulation. We will also compare the performance of the SIMPLE-LDG and SIMPLE-SIP method in terms of the number SIMPLE iterations, which are needed to reach the specified convergence criterion. The stability of the SIMPLE-LDG and SIMPLE-SIP method is studied by solving the Orr-Sommerfeld problem in Section 5.2.2 using the mixed- and equal-order formulation. Finally, we validate the scheme by simulating the flow past a square cylinder in Section 5.2.3.

5.2.1 Taylor vortex flow

To investigate the temporal and spatial accuracy as well as the performance of the SIMPLE algorithm we solve the Taylor vortex flow as the first unsteady test case. For this benchmark the analytical solution of the Navier-Stokes equations is given by

$$u_1 = -\cos(\pi x_1)\sin(\pi x_2)\exp\left(\frac{-2\pi^2 t}{Re}\right), \tag{5.9}$$

$$u_2 = \sin(\pi x_1)\cos(\pi x_2)\exp\left(\frac{-2\pi^2 t}{Re}\right), \tag{5.10}$$

$$p = -\frac{\cos(2\pi x_1)+\cos(2\pi x_2)}{4}\exp\left(\frac{-4\pi^2 t}{Re}\right). \tag{5.11}$$

The Reynolds number is set to $Re = 100$ and the computational domain $\Omega = [-1,1]^2$ is discretized by uniform Cartesian grids. On all domain boundaries we use periodic boundary conditions. Initial conditions are taken from the analytical solution (5.9)-(5.11) for $t = 0$. For the BDF schemes of order $s \geq 2$ additional start values are calculated for $t_{-1} = -\Delta t, \ldots, t_{-(s-1)} = -(s-1)\Delta t$. The results shown below are calculated for the time $t_E = 5$, where the initial velocity field is decayed to approximately one third. For the time t_E we measure the errors of the calculated velocity and pressure in the L^2-norm with respect to the exact solution. As convergence criterion for the SIMPLE iterations in each time step we set the L^2-norms of the pressure correction and the change in each velocity component to $\|p_h'\|_{L^2(\Omega)} \leq 10^{-12}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} \leq 10^{-12}$. To fix the constant offset in the calculated pressure the mean value is set to zero, i.e. $\int_\Omega p\, d\mathbf{x} = 0$. Next, we show the temporal and spatial convergence studies.

5.2.1.1 Temporal convergence study [1]

To measure the temporal convergence rates for the different BDF schemes of order 1 to 4 the time step size Δt is varied between 0.005 and 1. For the spatial discretization the mixed-order SIMPLE-LDG method is used. By taking a grid with 10×10 cells and polynomial orders of $k = 8$ for the velocity and $k' = 7$ for the pressure, it is assured that the temporal errors dominate the spatial ones. In Fig. 5.8a and b the errors for velocity and pressure are plotted against the time step size Δt. The temporal convergence rates are found to be in very good agreement with the values expected from the theory.

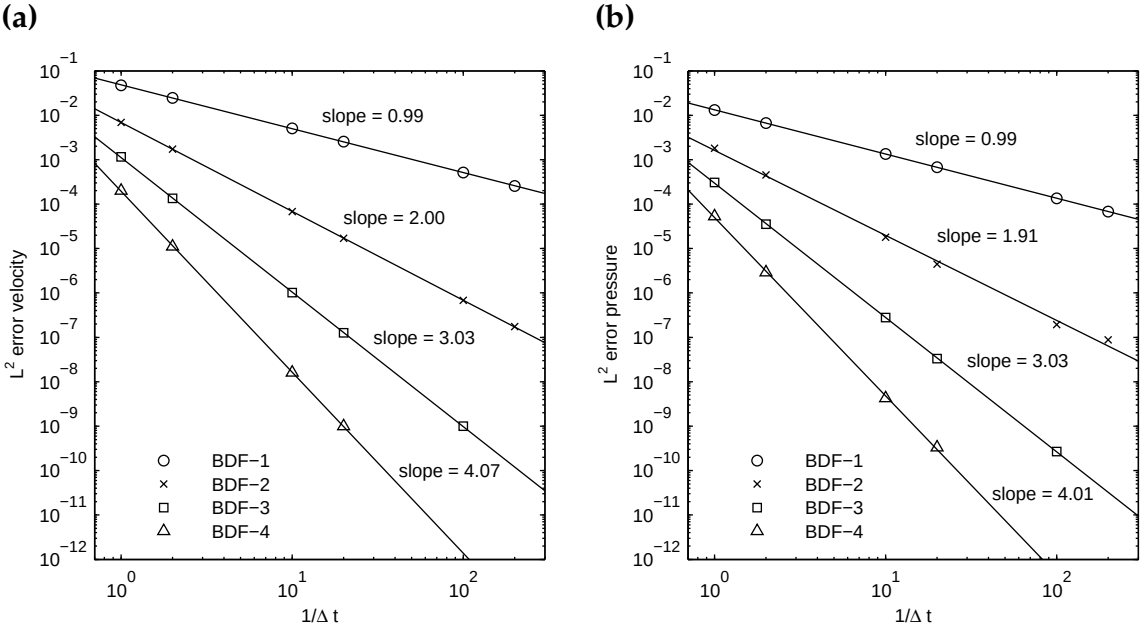


Figure 5.8: Temporal convergence study for the Taylor vortex flow using the mixed-order SIMPLE-LDG method: (a) L^2 error of velocity and (b) L^2 error of pressure vs. time step size. Also shown are the best linear fits in a least square sense and its slopes.

5.2.1.2 Spatial convergence study

For the spatial convergence study we take the BDF-4 scheme with a time step size $\Delta t = 0.01$, for which the spatial errors dominate the temporal errors. The polynomial orders for the velocity are $k = 1, \ldots, 5$ and for the pressure $k' = 0, \ldots, 4$ and $k' = 1, \ldots, 5$ for the mixed- and equal-order formulation, respectively. We observe that the errors of velocity and pressure in the L^2-norm for the SIMPLE-LDG and the SIMPLE-SIP method are virtually the same. Therefore, we only show the results for the SIMPLE-LDG method.

[1] Note that the simulations for the temporal convergence study for the Taylor vortex flow have been performed by (Keil 2012).

In Fig. 5.9 the p-convergence on a grid with 48×48 cells is shown. As expected, an exponential rate of convergence can clearly be observed, where the results using the equal-order formulation are slightly more accurate.

To test the h-convergence we take uniform Cartesian grids with 12×12 cells for the coarsest one and 48×48 cells for the finest one. In Fig. 5.10a and b the h-convergence of the velocity and the pressure for the mixed-order formulation is displayed. We observe the optimal convergence rates of approximately $k+1$ and $k'+1$ for velocity and pressure, respectively. For the equal-order formulation the results are shown in Fig. 5.10c and d. Again, the velocity converges with the optimal rate of approximately $k+1$. However, the observed convergence rates for the pressure are suboptimal, which is in agreement with the theoretical and numerical results in (Cockburn et al. 2009). The obtained convergence rate is k' for the polynomial orders $k'=1$ and $k'=5$. For the remaining polynomial orders the convergence rate for the pressure is slightly higher than k'.

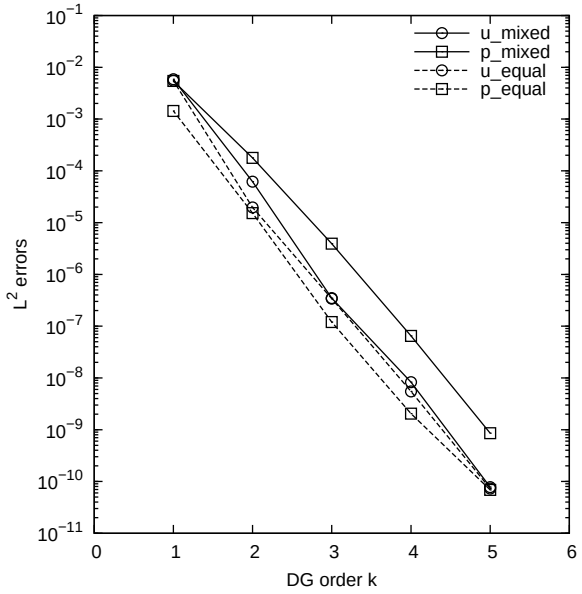


Figure 5.9: p-convergence study on a grid with 48×48 cells for the Taylor vortex flow using the SIMPLE-LDG method: L^2 errors of velocity and pressure vs. polynomial order. For the mixed-order formulation the polynomial degrees are k for velocity and $k-1$ for pressure.

5.2.1.3 Performance

In this section, we compare the performance of the SIMPLE-LDG and SIMPLE-SIP for the Taylor vortex flow using the mixed- and equal-order formulation. The performance is evaluated by comparing the averaged number of SIMPLE iterations for one time step to reach the specified convergence criterion, which is $\|p'_h\|_{L^2(\Omega)} \leq 10^{-12}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} \leq 10^{-12}$. The relaxation factors for velocity and pressure are set to

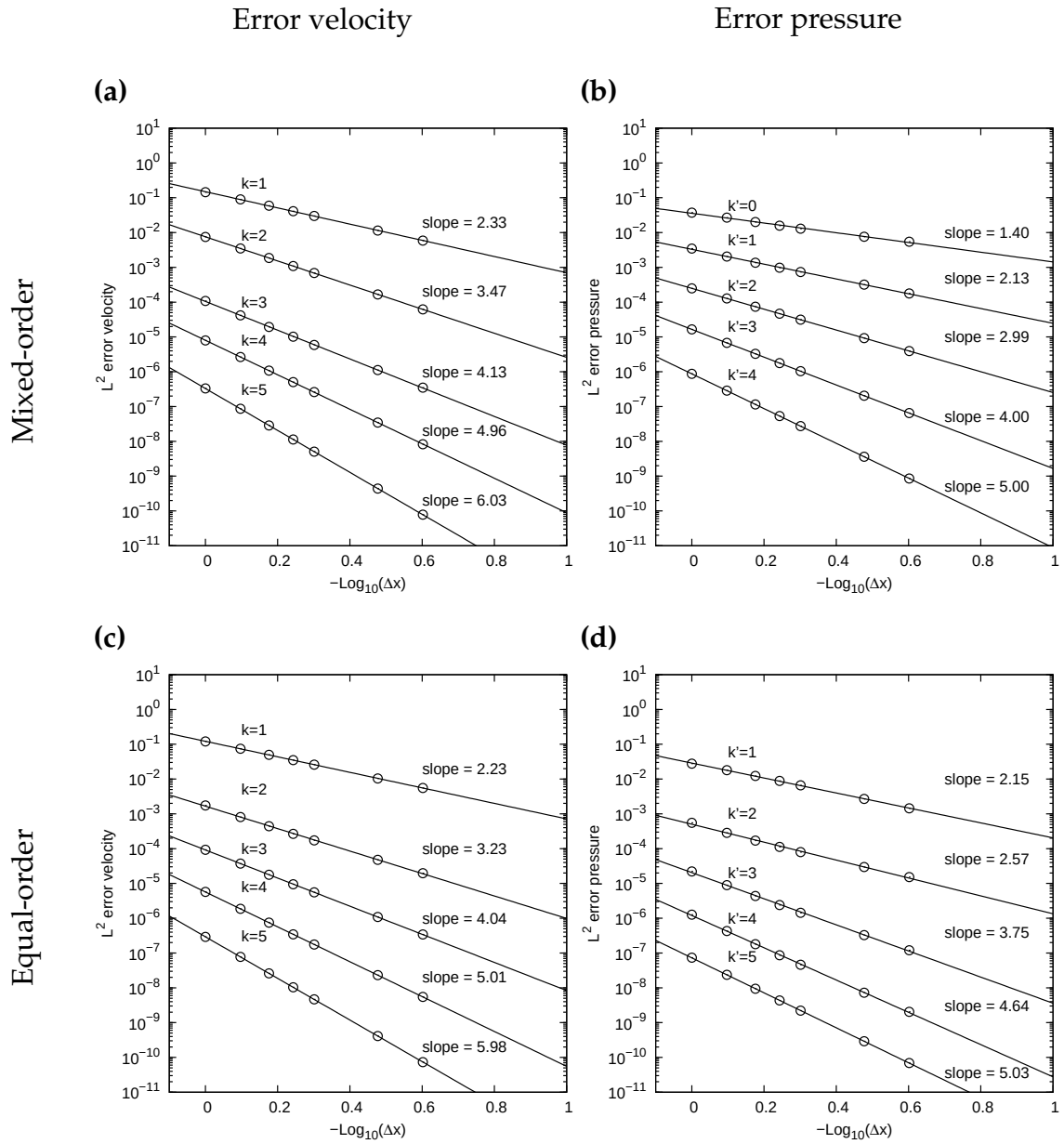


Figure 5.10: *h*-convergence study for the Taylor vortex flow using the SIMPLE-LDG method: (a) and (b) L^2 errors of velocity and pressure for the mixed-order formulation vs. grid size, (c) and (d) L^2 errors of velocity and pressure for the equal-order formulation vs. grid size. The grid size Δx is normalized with respect to the coarsest grid. Also shown are the best linear fits in a least square sense and its slopes.

$\alpha_u = 1.0$ and $\alpha_p = 1.0$ for all simulations. The averaged number of SIMPLE iterations is calculated by the total number of SIMPLE iterations divided by the number of time steps.

For the first performance test the initial guesses for the SIMPLE iterations in each time step for velocity and pressure are set to zero. The results for the mixed-order SIMPLE-LDG method are shown in Fig. 5.11a. In general, the number of SIMPLE iterations is increasing with finer grids and polynomials of higher order, which is to be expected for a segregated solution algorithm. The decrease in the number of SIMPLE iterations for the highest polynomial order $k = 5$ and $k' = 4$ considered can be understood by the results in Fig. 5.11b, where the convergence history of the pressure correction is plotted for the first time step on the finest grid for varying polynomial orders. One can observe that the speed of convergence for each polynomial order is not constant and hence the number of SIMPLE iterations for different polynomial orders is strongly influenced by the specified convergence criterion.

Next, we compare the performance of the SIMPLE-LDG and SIMPLE-SIP method. Therefore, we initialize the SIMPLE iterations in each time step with the solutions of velocity and pressure from the previous time step. As expected, this significantly improves the performance, which can be found by comparing Fig. 5.11a and Fig. 5.12a for the mixed-order SIMPLE-LDG method. In Fig. 5.12 the results for the mixed- and equal-order SIMPLE-LDG and SIMPLE-SIP method are shown. Overall, the mixed-order SIMPLE-LDG method performs best. At the same time, referring to Section 4.5.3, the mixed-order SIMPLE-LDG approach is the only method, which provides up to machine accuracy a divergence-free velocity field.

5.2.2 Orr-Sommerfeld stability problem

In the second unsteady test case, we study the numerical stability of our proposed scheme by calculating the evolution of small disturbances in a channel flow. Therefore, we solve the Orr-Sommerfeld problem, which examines the hydrodynamic stability and is investigated in many numerical studies, see e.g. (Shahbazi et al. 2007) in the context of the DGM and (Malik, Zang & Hussaini 1985) in the context of the Fourier-Chebyshev spectral method. The geometry for this test case is a two-dimensional channel $\Omega = [-\pi, \pi] \times [-1, 1]$ with the no-slip boundary condition at the lower and upper walls and periodic boundary conditions in the streamwise direction. From linear stability analysis by solving the Orr-Sommerfeld equation for plane Poiseuille flow superposed with small disturbances one obtains for the velocity field

$$u_1 = 1 - x_2^2 + \epsilon \hat{u}_1, \tag{5.12}$$

$$u_2 = \epsilon \hat{u}_2, \tag{5.13}$$

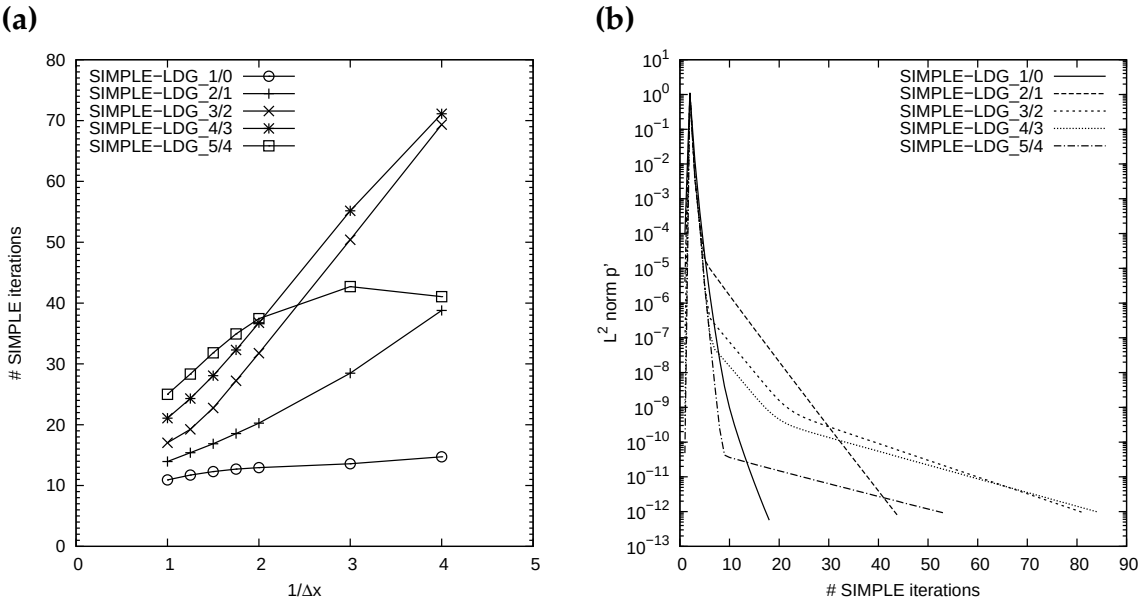


Figure 5.11: Performance of the SIMPLE algorithm for the Taylor vortex flow using the mixed-order SIMPLE-LDG method: (a) averaged number of SIMPLE iterations for one time step vs. grid size for different polynomial orders denoted by SIMPLE-LDG_k/k', (b) convergence history of the pressure correction vs. number of SIMPLE iterations for the first time step on the finest grid with 48×48 cells. The initial solutions for the SIMPLE iterations in each time step for both velocity components and the pressure are set to zero.

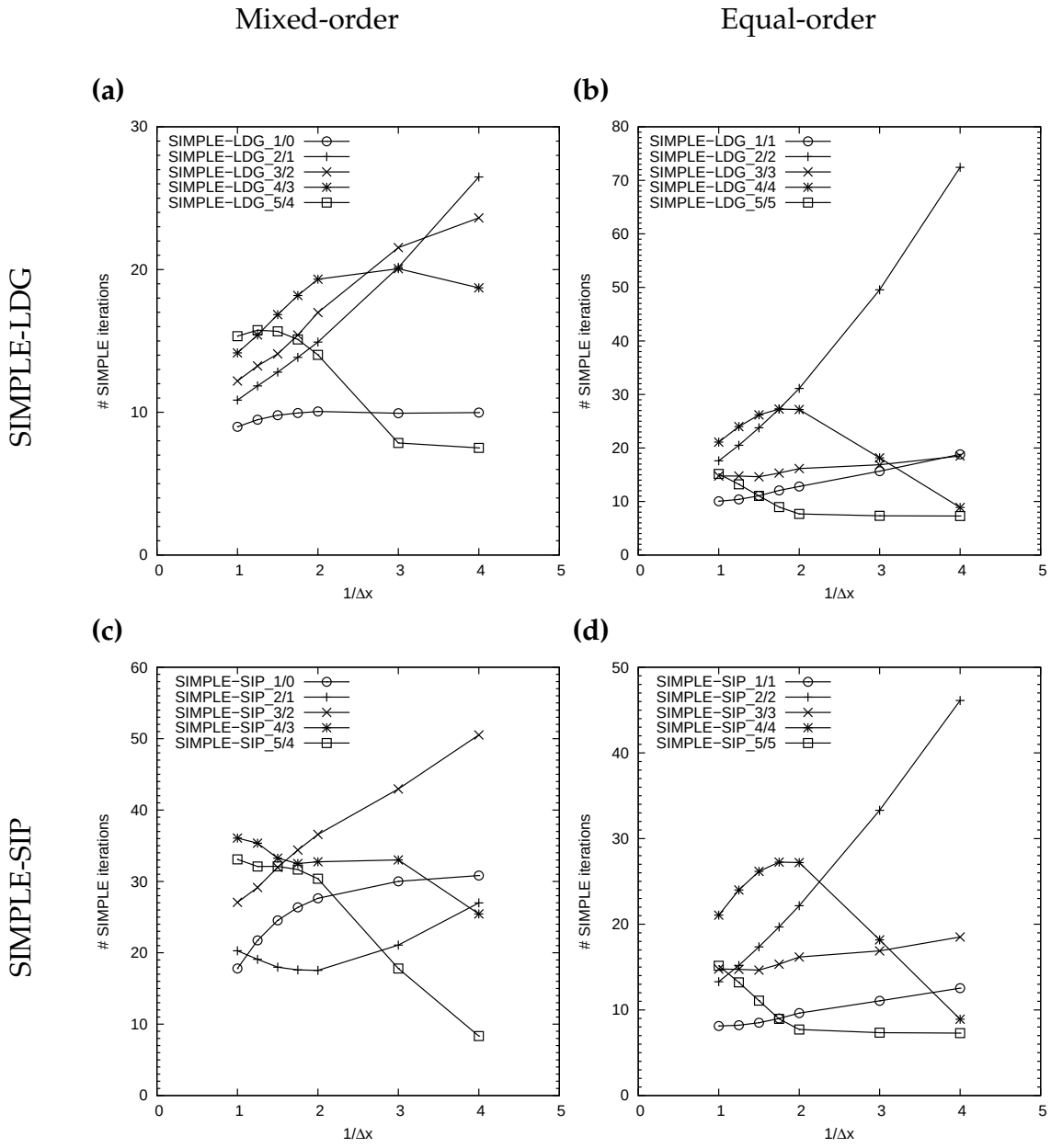


Figure 5.12: Performance of the SIMPLE algorithm for the Taylor vortex flow. The averaged number of SIMPLE iterations for one time step is plotted vs. the grid size for different polynomial orders: (a) and (b) mixed- and equal-order SIMPLE-LDG method, (c) and (d) mixed- and equal-order SIMPLE-SIP method. The SIMPLE iterations in each time step are initialized with the solutions of velocity and pressure from the previous time step.

where $\hat{u}_1$ and $\hat{u}_2$ are the disturbances and ϵ is an amplitude parameter. The disturbances have the form [2]

$$\hat{u}_1 = \Re\left\{\frac{d\psi(x_2)}{dx_2} e^{i(\alpha x_1 - \omega t)}\right\}, \tag{5.14}$$

$$\hat{u}_2 = -\Re\left\{i\alpha\psi(x_2) e^{i(\alpha x_1 - \omega t)}\right\}, \tag{5.15}$$

with $\psi(x_2)$ being a complex eigenfunction of the Orr-Sommerfeld equation

$$-\frac{1}{i\alpha Re}\psi'''' + \left(U + \frac{2\alpha^2}{i\alpha Re}\right)\psi'' + \left(-U\alpha^2 - U'' - \frac{\alpha^4}{i\alpha Re}\right)\psi = \frac{\omega}{\alpha}\left(\psi'' - \alpha^2\psi\right), \tag{5.16}$$

where U is the solution of the base flow, i.e. $U = 1 - x_2^2$ for plane Poiseuille flow. In (5.14)-(5.15) the wave number α has to be prescribed and ω/α is the corresponding complex eigenvalue to the eigenfunction ψ, which is normalized to a maximum value of 1, i.e. $\max|\psi| = 1$. As long as the disturbances are small compared to the mean flow, the perturbation energy

$$E(t) = \int_{-\pi}^{\pi}\int_{-1}^{1}\left(\left(1 - x_2^2 - u_1\right)^2 + u_2^2\right) dx_2\, dx_1 \tag{5.17}$$

should evolve in time with $E(t) = E(0)e^{2\omega_i t}$, where ω_i denotes the imaginary part of ω.

For our calculations we set up the parameters following (Shahbazi et al. 2007, Malik et al. 1985), i.e. $Re = 7500$ and $\alpha = 1$. Then, there is only one unstable eigensolution with $\omega_i = 0.002234976$. The amplitude parameter is set to $\epsilon = 10^{-4}$. In Fig. 5.13 the rectangular grid consisting of 8×20 cells, which is refined towards the lower and upper walls, is shown. In our studies a second refined grid with 16×40 cells is used. For the time discretization the BDF-2 scheme with a time step size of $\Delta t = 10^{-3}$ is used. The very first time step is calculated using the BDF-1 scheme and initial conditions are taken from (5.12)-(5.15) with $t = 0$. The SIMPLE iterations within one time step are terminated when the L^2-norms of the pressure correction $\|p_h'\|_{L^2(\Omega)}$ and the change in each velocity component $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)}$ become smaller than the specified tolerance of 10^{-12}. Due to the periodic boundary conditions in the streamwise direction a constant pressure gradient is applied in the x_1-direction and a reference point for the pressure has to be prescribed. By our numerical experiments we found, that for the equal-order formulation the penalty for the pressure stabilization heavily influences the performance of the SIMPLE algorithm. The performance is strongly improved by omitting the Re number in (4.14). The stability of the method is not influenced by that choice of the penalty for pressure stabilization.

The following results concerning the stability of the proposed scheme are the same for the SIMPLE-LDG and the SIMPLE-SIP method. Therefore, we show only the results for

[2] Note the algebraic sign for $\hat{u}_2$ in (5.15) was wrong in (Klein et al. 2015) and has been corrected in here.

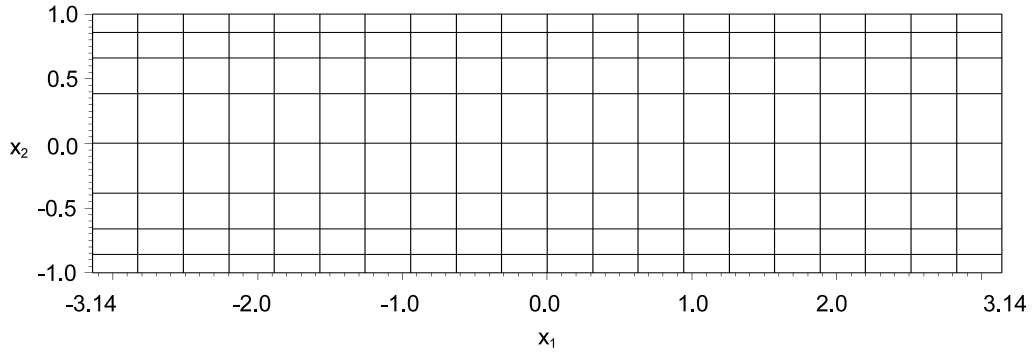


Figure 5.13: Geometry and computational grid consisting of 8×20 cells for the Orr-Sommerfeld stability problem.

the SIMPLE-LDG method. When using the mixed-order formulation, our numerical experiments have shown that the stability of the method is strongly dependent on the penalty parameter η of the interior penalty discretization for the diffusive operator as well as the mesh size. To investigate this effect different penalty parameters are tested $\eta^* = \eta_{\mathrm{Sc}}\eta$, where $\eta_{\mathrm{Sc}} \geq 1.0$ is a constant scaling factor and η follows from Eqs. (4.9), (4.10) and (4.12). The perturbation energy growth rate versus time for different scaling factors η_{Sc} with polynomial orders $k = 5$ for velocity and $k' = 4$ for pressure is plotted in Fig. 5.14a and 5.14b on grids with 8×20 and 16×40 cells, respectively. The results in Fig. 5.14a on the coarser grid show unphysical instabilities for $\eta_{\mathrm{Sc}} = 1.0$ and 2.0, which occur at a later time for increasing scaling factors. For $\eta_{\mathrm{Sc}} = 5.0$ the computation is stable and the results are very close to those from linear stability analysis. Reducing the polynomial order from $k = 5$, $k' = 4$ to $k = 4$, $k' = 3$ on the coarse grid we find the same scaling factor of $\eta_{\mathrm{Sc}} = 5.0$ for stability. On the finer grid a scaling factor of $\eta_{\mathrm{Sc}} = 2.0$ is sufficient to stabilize the simulations, cf. Fig. 5.14b. Increasing the scaling factor η_{Sc} further does not influence the results.

In Fig. 5.14c and 5.14d the same plots are shown for varying polynomial orders for the mixed- and equal-order formulation on the coarse grid. For the mixed-order formulation the scaling factor is set to $\eta_{\mathrm{Sc}} = 5.0$. The equal-order formulation is stable without rescaling the penalty of the interior penalty discretization for the diffusive operator, i.e. $\eta_{\mathrm{Sc}} = 1.0$. As expected, for increasing polynomial orders the results are getting closer to those obtained by linear stability analysis and the equal-order formulation is slightly more accurate than the mixed-order formulation.

For the mixed-order formulation we further test the long-term stability for the polynomial degrees $k = 5$ and $k' = 4$ with $\Delta t = 10^{-2}$ and $\eta_{\mathrm{Sc}} = 5.0$ on the coarse grid. Let $T_0 = 25.1436$ be the time for the perturbation waves of length 2π to travel once through the channel. Then, the perturbation energy growth rate is in the range of the one from linear stability analysis until approximately $t/T_0 = 70$, where non-linear effects come into play. Finally, we examine the spatial accuracy for the case $\eta_{\mathrm{Sc}} = 5.0$

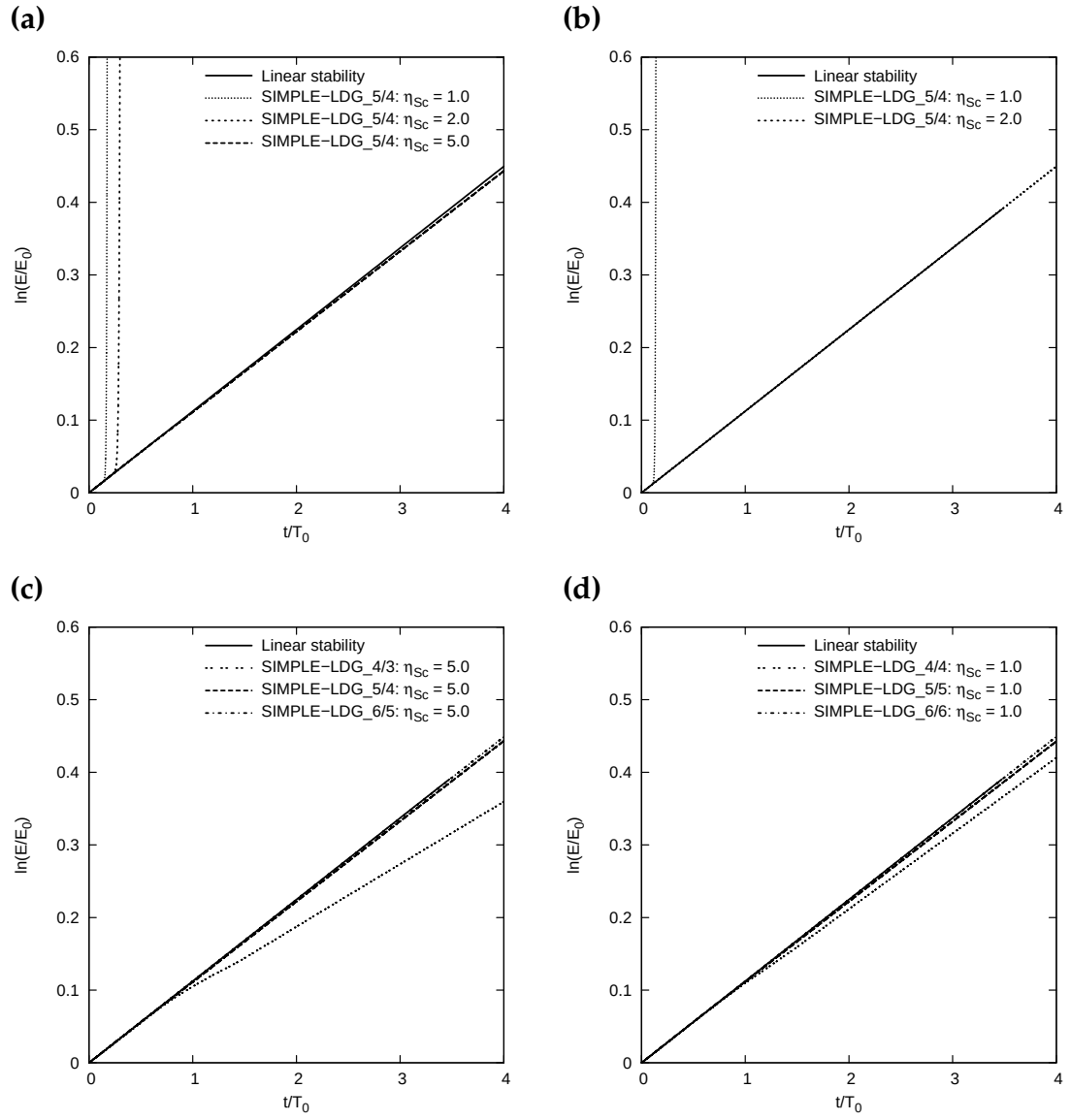


Figure 5.14: Perturbation energy growth rate vs. normalized time for the Orr-Sommerfeld stability problem. SIMPLE-LDG_k/k' denotes the solution of the Navier-Stokes equations computed with the unsteady SIMPLE algorithm using the original formulation for the pressure correction. The polynomial orders are k for velocity and k' for pressure. The energy is normalized with $E_0 = E(0)$ and the time with $T_0 = 25.1436$, the period for the perturbation waves to travel once through the channel. (a) and (b): Results for different scaling factors η_{Sc} for the penalty parameter of the diffusive operator on grids with 8×20 and 16×40 cells, respectively. (c) and (d): Results for the mixed- and equal-order formulation for different polynomial orders on a grid with 8×20 cells.

on the coarse grid. For this purpose, we define the error in the perturbation energy growth rate

$$e_g(t) = \frac{1}{\omega_i} \left| \omega_i - \frac{1}{2t} \ln \left(\frac{E(t)}{E(0)} \right) \right|, \tag{5.18}$$

where ω_i is the result from linear stability analysis. The error for time $t = 60$ is plotted against the polynomial order of the velocity for $k = 4 - 7$ in Fig. 5.15. A spectral rate of convergence can clearly be observed.

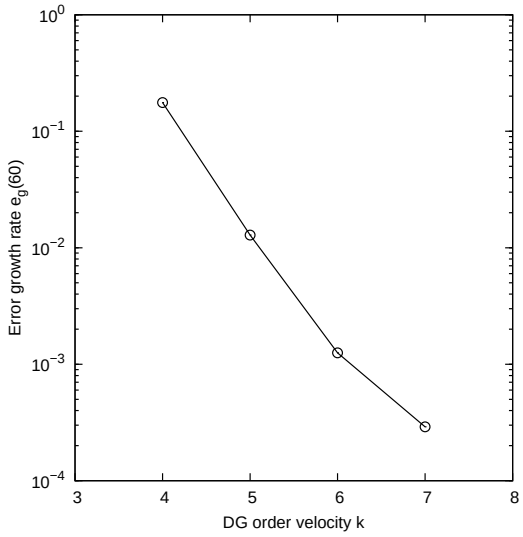


Figure 5.15: Error in the perturbation energy growth rate vs. polynomial order of the velocity for the Orr-Sommerfeld stability problem using the mixed-order SIMPLE-LDG method on a grid with 8×20 cells and the scaling factor for the penalty parameter of the diffusive operator $\eta_{Sc} = 5.0$. Polynomial orders of the pressure are $k' = k - 1$.

5.2.3 Flow past a square cylinder

In this section, the proposed scheme for unsteady incompressible flows is validated by simulating the flow past a square cylinder. First, we show in Section 5.2.3.1 results for two-dimensional flow at $Re = 100$, where the characteristic vortex shedding of the well-known von Kármán vortex street can be observed. Then, in Section 5.2.3.2 we investigate the critical Reynolds numbers for the onset of vortex shedding and the transition to three-dimensional flow.

5.2.3.1 Two-dimensional vortex shedding

For the flow past a square cylinder we use the mixed-order SIMPLE-LDG method, since this method is the most efficient in terms of SIMPLE iterations and has got approximately the same accuracy like the equal-order method as shown for the first

two test cases. The computational domain is $\Omega = [-12.5, 50] \times [-12.5, 12.5]$. The center of the square cylinder with the edge length $L = 1$ is placed at $(x_C, y_C) = (0, 0)$ resulting in a blockage ratio of $B = 4\%$, which is defined as the ratio of the edge length of the cylinder to the height of the computational domain. In Fig. 5.16 the computational grid with 1610 triangles is shown. The grid is refined towards the walls of the cylinder as well as in the wake of the cylinder. At the left domain boundary an inflow condition (i.e. Dirichlet boundary condition) with a uniform velocity profile is applied $u_1(x_1 = -12.5, x_2) = U_\infty$ and $u_2(x_1 = -12.5, x_2) = 0$, where $U_\infty = 1$. At the remaining outer domain boundaries we use the outflow boundary condition (2.4b) and at the walls of the cylinder the no-slip condition is employed. The Reynolds number is set to $Re = 100$.

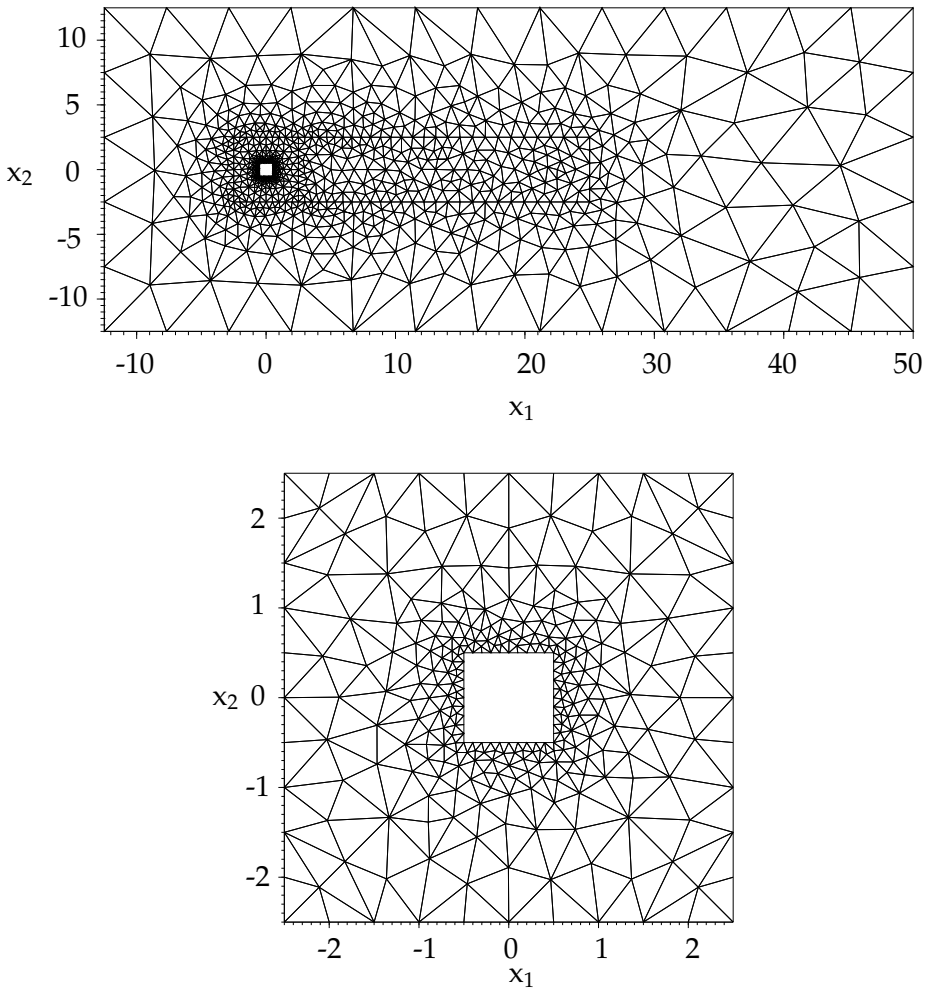


Figure 5.16: Computational grid with 1610 cells for the flow past a square cylinder (top). Zoom-in of the grid in the region close to the cylinder (bottom).

The simulation is performed with polynomial orders of $k = 3$ and $k' = 2$ for velocity and pressure, respectively. The time step size is $\Delta t = 0.1$ using the BDF-2 scheme. Again, the very first time step is calculated using the BDF-1 scheme, where the initial velocity field is $u_1(t = 0) = 1$ and $u_2(t = 0) = 0$. As convergence criterion for the

SIMPLE iterations in each time step we use $\|p'_h\|_{L^2(\Omega)} \leq 10^{-6}$ and $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} \leq 10^{-6}$. Below the results are shown after reaching a time-periodic state.

Comparisons between numerical results and experiments for this test case are usually based on the non-dimensional Strouhal number $St = fL/U_\infty$, where f is the frequency of the total lift coefficient. We calculated the Strouhal number for our test to 0.144, which is in very good agreement with experiments (Okajima 1982) and numerical computations (Darekar & Sherwin 2001, Shahbazi et al. 2007) as shown in Table 5.5. A contour plot of the instantaneous vorticity field is shown in Fig. 5.17a, where the well-known von Kármán vortex street can be observed. In Fig. 5.17b and c the total lift and drag coefficients are shown over a time span of $t' \in [0, 50]$ after reaching the time-periodic state. The values for the root mean square of the total lift coefficient and the mean of the total drag coefficient [3] are calculated to $C'_L = 0.186$ and $\overline{C_D} = 1.488$, which is in perfect agreement with the values reported in (Darekar & Sherwin 2001).

Table 5.5: Comparison of experimental data and numerical results of the Strouhal number for flow past a square cylinder at $Re = 100$. B indicates the blockage ratio.

Reference	St
Okajima (1982) (experimental), $B = 0\%$	0.141-0.145
Darekar & Sherwin (2001), $B = 4.2\%$	0.146
Shahbazi et al. (2007), $B = 2.3\%$	0.145
Present, $B = 4\%$	0.144

Finally, in Fig. 5.18 the convergence history of the SIMPLE algorithm is displayed. The pressure correction and the changes in each velocity component are plotted versus the number of SIMPLE iterations for 10 time steps in the time-periodic region of the solution. A drop of about 10^4 for the pressure correction and of 10^6 for the velocity components is reached in each time step. On overall average, about 40 SIMPLE iterations are performed each time step.

5.2.3.2 Critical Reynolds numbers for onset of vortex shedding and transition to three-dimensional flow

During the course of this thesis, in (Fischer 2014) the critical Reynolds numbers for the onset of vortex shedding and the transition from two- to three-dimensional flow were investigated. The main results are summarized here. For small Reynolds numbers the flow is two-dimensional, steady and symmetric with respect to the horizontal axis. Increasing the Reynolds number the flow is getting unsteady and exhibits the von Kármán vortex street, like shown in the previous section for $Re = 100$. At this point,

[3] $C'_L = \left(\overline{C_L^2}\right)^{1/2} = \left(\sum_{n=0}^{N} (C_L(t'_n))^2/(N+1)\right)^{1/2}$ and $\overline{C_D} = \sum_{n=0}^{N} C_D(t'_n)/(N+1)$, where N is the number of time steps for $t'_n \in [0, 50]$.

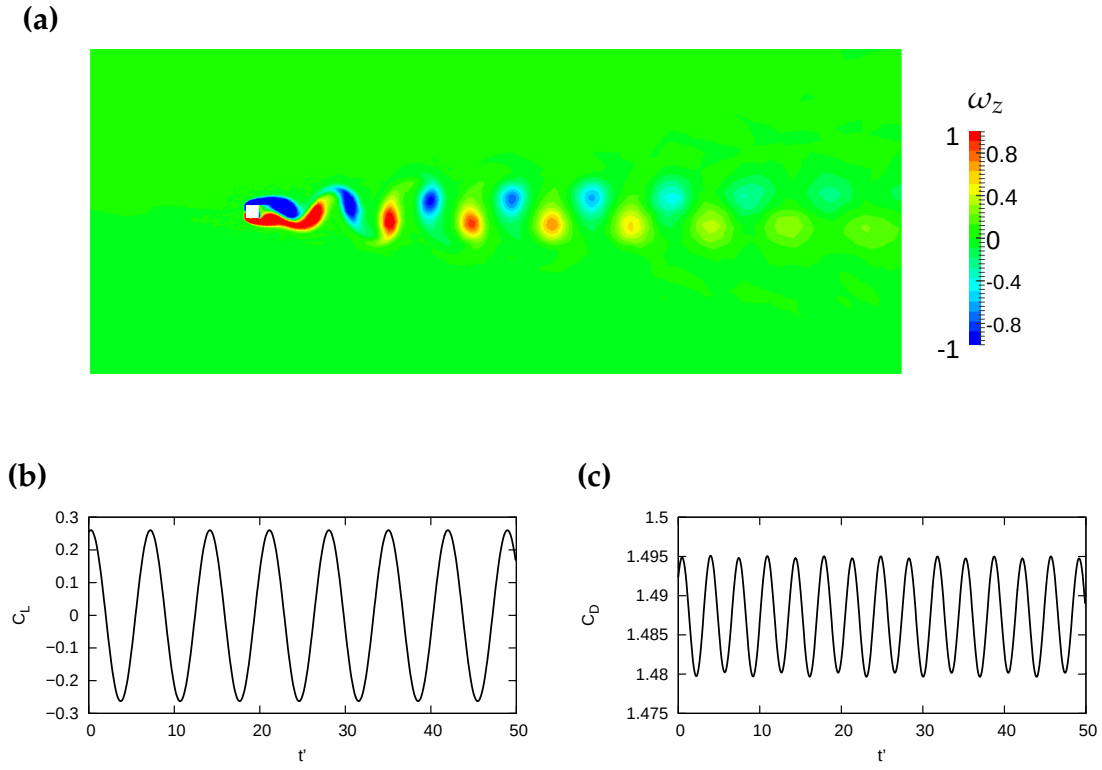


Figure 5.17: Flow past a square cylinder at $Re = 100$. (a) Contour plot of instantaneous vorticity field. (b) Total lift coefficient and (c) total drag coefficient over time for a time span $t' = 0 \ldots 50$ after reaching a time-periodic state.

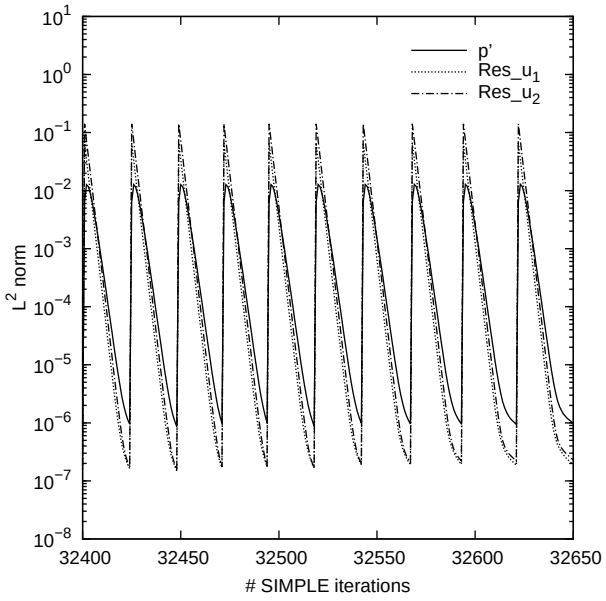


Figure 5.18: Performance of the SIMPLE algorithm for flow past a square cylinder at $Re = 100$. Pressure correction and changes in velocity components versus number of SIMPLE iterations for 10 time steps in the time-periodic region of the solution.

the flow is still two-dimensional. The transition from steady two-dimensional flow to the onset of vortex shedding is characterized by the first critical Reynolds number Re_{cr1}. Using the unsteady SIMPLE algorithm for incompressible flows developed in this thesis the critical value for Re_{cr1} was determined for two different domains $[-9.5, 12.5] \times [-9.0, 9.0]$ with 1771 cells and $[-6.0, 18.0] \times [-5.0, 5.0]$ with 1780 cells. The first one yields a blockage ratio of $B = 6\%$ and the second one of $B = 10\%$. In both cases rectangular grids were used, which are refined towards the cylinder surface. To observe the onset of vortex shedding the cylinder had to be slightly shifted in the vertical direction, i.e. the cylinder must not be in the center of the domain. For the first domain with $B = 6\%$ the critical Reynolds number was determined at $Re_{cr1} = 47 \pm 1$ and for the second domain with $B = 10\%$ at $Re_{cr1} = 43 \pm 1$. It is generally accepted that the critical values depend on the blockage ratio as well as the domain size in the streamwise direction. The found numbers agree very well with the reference values from literature, which can be found in (Fischer 2014).

In a next step, three-dimensional simulations were performed to determine the second critical Reynolds number Re_{cr2}, which is defined by the transition from two- to three-dimensional flow. Again, the cylinder was slightly shifted in the vertical direction. An initial value for the velocity in the spanwise direction had to be set $u_3(t = 0) = 0.1$ to numerically trigger the three-dimensional flow. For the domain $[-6.0, 25.0] \times [-9.0, 9.0] \times [-3.0, 3.0]$ with 32410 cells the critical value was identified to be in the range of $Re_{cr2} = 180 - 185$. In Fig. 5.19 instantaneous contours of the spanwise vorticity contours are shown for $Re = 180$. It can clearly be seen that the contours are constant in the spanwise direction. The vorticity in the streamwise as well as in the cross-stream direction are still zero indicating the two-dimensional character of the flow at $Re = 180$. The instantaneous vorticity contours for $Re = 185$ can be seen in Fig. 5.20. Evidently, the flow is three-dimensional. Compared with values in the literature for Re_{cr2} the range from 180 to 185, which we found, is slightly higher. However, there is no exact value in the literature for this critical Reynolds number and only ranges are provided. And again, the critical value depends on many aspects like domain size, blockage ratio and boundary conditions. But we can state that the physical phenomena observed below and above the critical value are the same like reported in the literature, cf. (Fischer 2014).

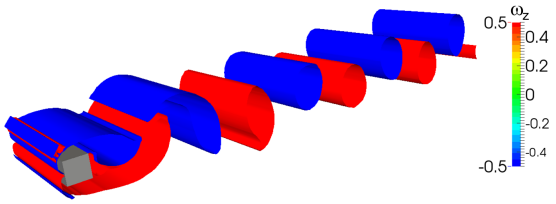


Figure 5.19: Instantaneous contours of spanwise vorticity for three-dimensional flow past a square cylinder at $Re = 180$.

A third critical Reynolds number Re_{cr3} can be defined by inspecting the wake structure behind the cylinder and the course of the lift and drag force at the cylinder over time.

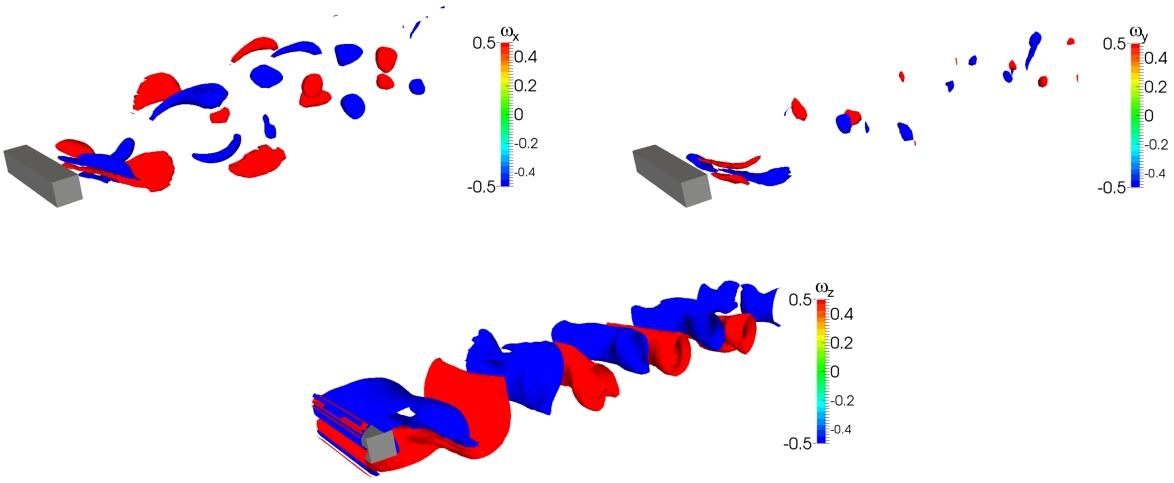


Figure 5.20: Instantaneous contours of streamwise vorticity (top left), cross-stream vorticity (top right) and spanwise vorticity (bottom) for three-dimensional flow past a square cylinder at $Re = 185$.

Below Re_{cr3} there is a regular pulsation of the lift and drag force, which is vanishing for higher Reynolds numbers. In (Fischer 2014) this third critical Reynolds number was identified to be between 230 and 240. The instantaneous vorticity contours for $Re = 240$ are displayed in Fig. 5.21, showing that the flow structure is getting more complex. Also this critical value is higher compared to those found in literature, which can be explained by the same reasons like already discussed above. Finally, in Fig. 5.22 vorticity contours for $Re = 300$ are shown, demonstrating that the flow field is getting more and more irregular compared to the lower Reynolds numbers. The results discussed above show evidence, that the implementation for unsteady incompressible flows is also correct for three-dimensional flows and can be considered as verified.

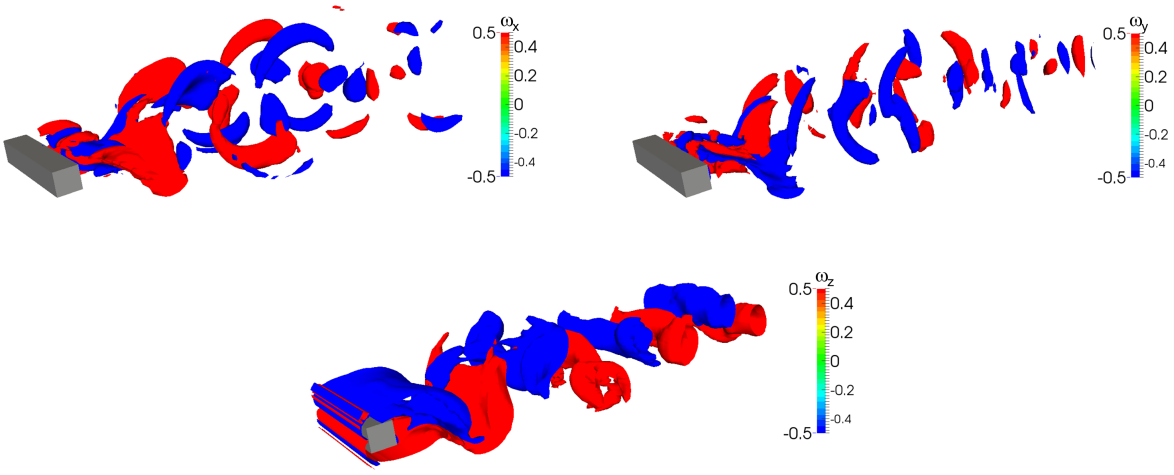


Figure 5.21: Instantaneous contours of streamwise vorticity (top left), cross-stream vorticity (top right) and spanwise vorticity (bottom) for three-dimensional flow past a square cylinder at $Re = 240$.

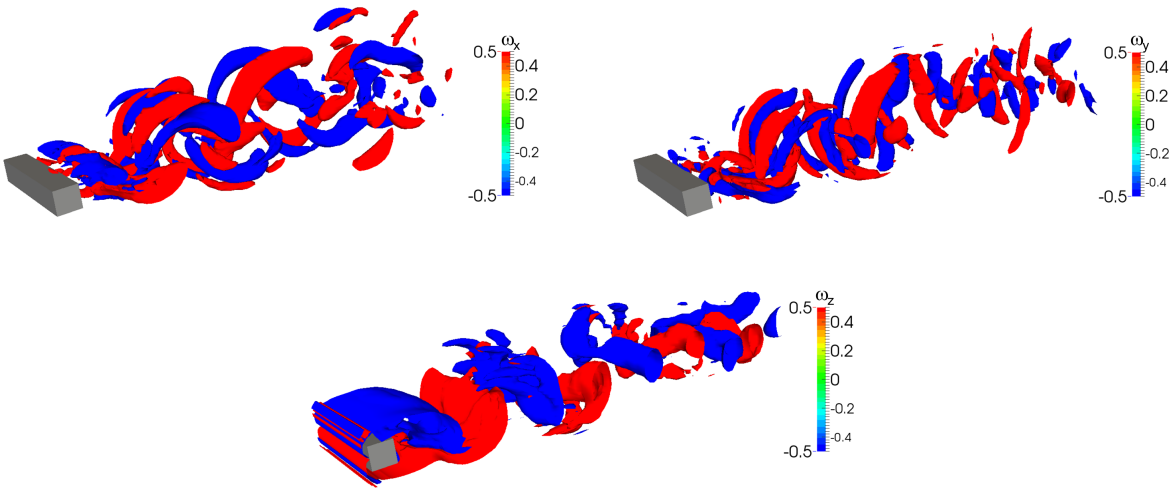


Figure 5.22: Instantaneous contours of streamwise vorticity (top left), cross-stream vorticity (top right) and spanwise vorticity (bottom) for three-dimensional flow past a square cylinder at $Re = 300$.

6 Discretization and algorithm for variable density flows

This chapter is devoted to the discretization and the algorithm for variable density flows, i.e. multiphase flows and low-Mach number flows. In Section 6.1 we review the literature on the application of the DGM to multiphase and low-Mach number flows. Then, in Section 6.2 we discuss the spatial discretization, which we applied in this work. The temporal discretization is given in Section 6.3. The implementation of jump conditions for multiphase flows in the DGM framework is elaborated in Section 6.4. Finally, in Section 6.5 the SIMPLE algorithm for variable density flows is derived.

6.1 Review of Discontinuous Galerkin solvers

In this section we give a review about Discontinuous Galerkin solvers for multiphase flows, cf. Section 6.1.1, and low-Mach number flows, cf. Section 6.1.2. We also comment on the approaches we are using.

6.1.1 Multiphase flows

There is very few literature about the application of the DGM to incompressible multiphase flows. In (Grooss & Hesthaven 2006) the level set method is used to capture the interface location and a smooth Heaviside function is applied for the transition between the two phases. The equations describing the flow field and the advection for the level set are solved in a segregated manner. The level set method with a smooth Heaviside
a fully coupled manner. In (Owkes & Desjardins 2013) the DGM is applied to solve the conservative level set, which is then coupled with a finite difference solver for the flow field. A smooth interface approach based on the level set method was also used in (Mousavi 2014) and (Emamy 2014), which was coupled with a projection scheme to solve the flow field. More recently, the development of sharp interface models based on the DGM has been started, see (Heimann et al. 2013) and (Kummer 2013).

In this work we use a conservative level set function, which was originally developed in (Olsson & Kreiss 2005) and (Olsson et al. 2007). The implementation of the multiphase solver in here was an intermediate step during the development of the low-Mach number solver. It does not contain surface tension nor the compressive-diffusive term of the conservative level set method to guarantee constant interface thickness. Anyway, these terms could be easily incorporated in the current implementation.

6.1.2 Low-Mach number flows

For simulating variable density flows, where the Mach number is small, there exist various continuum mechanical models, which can be used. For natural convection flows, where the temperature variation is small, one may apply the Boussinesq approximation. The incompressible Navier-Stokes equations are extended by a gravity term, which accounts for the variable density, and coupled with a transport equation for the temperature. In (Bassi & Crivellini 2006) and (Bassi et al. 2007) the DGM was applied to the Boussinesq approximation for solving natural convection problems. Like in (Bassi et al. 2006) an artificial compressibility flux is used for the transport term and the time integration is fully implicit.

Another obvious possibility is to solve the compressible Euler or Navier-Stokes equations, which is numerically a challenging task for low-Mach numbers. The smaller the Mach number the higher the difference between the speed of sound and the flow velocity, which in turn increases the stiffness of the system. Nevertheless, in (Luo, Baum & Löhner 2006) a p-Multigrid DGM was successfully used to solve the compressible Euler equations for flow past a circular cylinder at Mach numbers of $M = 0.01$ and $M = 0.38$. The equations were iterated to steady state using an explicit smoother on the higher polynomial levels and an implicit smoother on the lowest polynomial level. Also in (Feistauer & Kučera 2007) a semi-implicit DGM is used to solve the compressible Euler equations for irrotational flow past an airfoil and a circular half-cylinder at $M = 10^{-4}$.

A DGM for the compressible Navier-Stokes equations is developed in (Persson & Peraire 2008). Implicit time integration is used and Newton's method is applied to solve the non-linear system in each time step. The linear system in each Newton step is solved by the GMRES method. The performance of the GMRES method is investigated in dependence of the Mach number in the range between $M \approx 10^{-3}$ and $M \approx 10^{-1}$. While in general the convergence is decreasing for lower Mach number, this behavior could be improved by suitable preconditioning. In a series of papers (Bassi, De Bartolo, Hartmann & Nigro 2009), (Nigro, De Bartolo, Hartmann & Bassi 2010) and (Nigro, Renda, De Bartolo, Hartmann & Bassi 2013), a Discontinuous Galerkin solver for low Mach numbers based on the compressible Navier-Stokes equations is developed. To decrease the stiffness of the system for low Mach numbers preconditioning techniques are used. Note that preconditioning in this sense modifies the PDEs itself. The full preconditioning approach, which modifies the unsteady term as well as the numerical flux function, is compared with the flux preconditioning technique.

Also turbulent flows at low Mach numbers have been simulated applying the DGM to the compressible Navier-Stokes equations. In (Wei & Pollard 2011), direct numerical simulations (DNS) of turbulent channel flow with isothermal walls for various Mach numbers (the smallest one is $M = 0.2$) are performed. In (Covello, Nigro, De Bartolo & Florio 2014) and (Renda, Hartmann, De Bartolo & Wallraff 2015) the compressible Reynolds-averaged Navier-Stokes (RANS) equations coupled with the $k - \omega$ turbulence model are solved.

To the best of our knowledge, in this work it is the first time that the DGM is applied to solve the low-Mach number equations (2.14a)-(2.14d), which allows to simulate variable density flows at low Mach numbers using similar discretization techniques and solution strategies as for the incompressible case.

6.2 Spatial discretization

In this section, we discuss the spatial discretization of (2.8a)-(2.8c) for multiphase flows and (2.14b)-(2.14d) for low-Mach number flows. For ease of notation the remainder of this chapter (except Section 6.4) is based on the low-Mach number equations. Modifications for multiphase flows will be mentioned below.

6.2.1 Semidiscrete system of equations

We begin by deriving the semidiscrete system of equations of the unsteady low-Mach number equations (2.14b)-(2.14d) subjected to the initial conditions (2.20a)-(2.20b) and boundary conditions (2.19a)-(2.19e). The velocity and the temperature are approximated by $u_{hi} \in V_k$ and $T_h \in V_k$, respectively, and the pressure is approximated by $p_h \in V_{k'}$. We only consider the mixed-order formulation, i.e. $k' = k - 1$.

The spatial discretization for the momentum equation (2.14c) is derived by multiplying with the test functions $v^{l,m} \in V_k$, replacing u_i with its DG approximation u_{hi} and integrating over element K^l

$$\left\langle \frac{\partial \rho_h u_{hi}}{\partial t}, v^{l,m} \right\rangle = \int_{K^l} \frac{\partial \rho_h u_{hi}}{\partial t} v^{l,m} d\mathbf{x} =: \left[\frac{\partial}{\partial t} \left(\mathbf{M}_\rho(\underline{\tilde{T}}) \underline{\tilde{u}}_i \right) \right]_{(l,m)}, \tag{6.1a}$$

$$\begin{aligned} \left\langle \frac{\partial \rho_h u_{hi} u_{hj}}{\partial x_j}, v^{l,m} \right\rangle \quad &\approx \int_{\partial K^l} \hat{F}^C_{ij} n_{ej} v^{l,m} ds \\ &- \int_{K^l} \rho_h u_{hi} u_{hj} \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_C(\underline{\tilde{\rho}} \underline{\tilde{u}}) \underline{\tilde{u}}_i + \underline{b}_{C_i} \right]_{(l,m)}, \end{aligned} \tag{6.1b}$$

$$\begin{aligned} \left\langle \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_{hi}}{\partial x_j} \right), v^{l,m} \right\rangle \quad &\approx \int_{\partial K^l} \hat{F}^{D1}_i ds \\ &- \int_{K^l} \mu \frac{\partial u_{hi}}{\partial x_j} \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_{D1}(\underline{\tilde{\mu}}) \underline{\tilde{u}}_i + \underline{b}_{D1_i} \right]_{(l,m)}, \end{aligned} \tag{6.1c}$$

$$\left\langle \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_{hj}}{\partial x_i} \right), v^{l,m}_i \right\rangle \quad \approx \int_{\partial K^l} \hat{F}^{D2}_i ds$$

$$- \int_{K^l} \mu \frac{\partial u_{hj}}{\partial x_i} \frac{\partial v_i^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_{D2_{ij}}(\underline{\tilde{\mu}}) \underline{\tilde{u}}_j + \underline{b}_{D2_i} \right]_{(l,m)}, \quad (6.1d)$$

$$\left\langle \frac{\partial}{\partial x_i} \left(\mu \frac{\partial u_{hj}}{\partial x_j} \right), v_i^{l,m} \right\rangle \approx \int_{\partial K^l} \hat{F}_i^{D3} ds - \int_{K^l} \mu \frac{\partial u_{hj}}{\partial x_j} \frac{\partial v_i^{l,m}}{\partial x_i} d\mathbf{x} =: \left[\mathbf{A}_{D3_{ij}}(\underline{\tilde{\mu}}) \underline{\tilde{u}}_j + \underline{b}_{D3_i} \right]_{(l,m)}, \quad (6.1e)$$

$$\left\langle \frac{\partial p_h}{\partial x_i}, v^{l,m} \right\rangle \approx \int_{\partial K^l} \hat{F}^P n_{ei} v^{l,m} ds - \int_{K^l} p_h \frac{\partial v^{l,m}}{\partial x_i} d\mathbf{x} =: \left[\mathbf{A}_{P_i} \underline{\tilde{p}} + \underline{b}_{P_i} \right]_{(l,m)}, \quad (6.1f)$$

$$\left\langle \frac{1}{Fr^2} \rho_h \delta_{i2}, v^{l,m} \right\rangle = \int_{K^l} \frac{1}{Fr^2} \rho_h \delta_{i2} v^{l,m} d\mathbf{x} =: \left[\underline{b}_{f_i} \right]_{(l,m)}. \quad (6.1g)$$

The matrix $\mathbf{M}_\rho$ in (6.1a) is a block diagonal matrix with N blocks, where the dimension of each block is $N_k \times N_k$. For the derivation of the diffusive terms in (6.13) and (6.14) vector valued test functions $v_i^{l,m} \in (V_k)^d$ are used, since these terms couple the equations for the single velocity components u_{hi}. Within the linearization in the SIMPLE algorithm these operators will be split to decouple the equations for a segregated solution procedure, cf. Section 6.5. Also note that the matrices $\mathbf{M}_\rho(\underline{\tilde{T}})$, $\mathbf{A}_C(\underline{\tilde{\rho}}\underline{\tilde{u}})$, $\mathbf{A}_{D1}(\underline{\tilde{\mu}})$, $\mathbf{A}_{D2_{ij}}(\underline{\tilde{\mu}})$ and $\mathbf{A}_{D3_{ij}}(\underline{\tilde{\mu}})$ depend on some unknowns, i.e. these matrices need to be updated in each iteration during the solution procedure. For multiphase flows the diffusive operator (6.1e) will be zero, since the velocity field is divergence-free in that case. In (6.1f) and in the remainder of this work the index for the hydrodynamic pressure p_2 is dropped. We still keep the index for the thermodynamic pressure p_0.

Next, the continuity equation (2.14b) is discretized in space. Since we use a mixed-order formulation, the test functions for the continuity equation are $q^{l,m'} \in V_{k'}$ leading to

$$\left\langle \frac{\partial \rho_h}{\partial t}, q^{l,m'} \right\rangle = \int_{K^l} \frac{\partial \rho_h}{\partial t} q^{l,m'} d\mathbf{x} = \left[\frac{\partial \tilde{\rho}}{\partial t} \mathbf{I} \right]_{(l,m')}, \quad (6.2a)$$

$$\left\langle \frac{\partial \rho_h u_{hj}}{\partial x_j}, q^{l,m'} \right\rangle \approx \int_{\partial K^l} \hat{F}_j^{\mathrm{Div}} n_{ej} q^{l,m'} ds - \int_{K^l} \rho_h u_{hj} \frac{\partial q^{l,m'}}{\partial x_j} d\mathbf{x} =: \left[\sum_{j=1}^{d} \left(\mathbf{B}_j(\underline{\tilde{\rho}}) \underline{\tilde{u}}_j + \underline{b}_{B_j} \right) \right]_{(l,m')}, \quad (6.2b)$$

where the matrix $\mathbf{B}_j(\underline{\tilde{\rho}})$ depends on the density and hence will also need to be updated each SIMPLE iteration. Note that for multiphase flows the continuity equation (2.8a)

is the same like for incompressible flows. In that case, the continuity equation is discretized by (4.1e).

Finally, we need to discretize the temperature equation (2.14d) in space. The polynomial order for the temperature is equal to the order of the velocity, i.e. the test functions for the temperature equation are $v^{l,m} \in V_k$. Then, we get the following operators for the discrete temperature equation

$$\left\langle \frac{\partial \rho_h T_h}{\partial t}, v^{l,m} \right\rangle = \int_{K^l} \frac{\partial \rho_h T_h}{\partial t} v^{l,m} d\mathbf{x} =: \left[\frac{\partial}{\partial t} \left(\mathbf{M}_\rho(\underline{\tilde{T}}) \underline{\tilde{T}} \right) \right]_{(l,m)}, \tag{6.3a}$$

$$\begin{aligned} \left\langle \frac{\partial \rho_h u_{hj} T_h}{\partial x_j}, v^{l,m} \right\rangle \quad &\approx \int_{\partial K^l} \hat{F}_j^{CT} n_{ej} v^{l,m} ds \\ &- \int_{K^l} \rho_h u_{hj} T_h \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_{CT}(\underline{\tilde{\rho}} \underline{\tilde{u}}) \underline{\tilde{T}} + \underline{b}_{CT} \right]_{(l,m)}, \end{aligned} \tag{6.3b}$$

$$\begin{aligned} \left\langle \frac{\partial}{\partial x_j} \left(\lambda \frac{\partial T_h}{\partial x_j} \right), v^{l,m} \right\rangle \quad &\approx \int_{\partial K^l} \hat{F}^{DT} ds \\ &- \int_{K^l} \lambda \frac{\partial T_h}{\partial x_j} \frac{\partial v^{l,m}}{\partial x_j} d\mathbf{x} =: \left[\mathbf{A}_{DT}(\underline{\tilde{\lambda}}) \underline{\tilde{T}} + \underline{b}_{DT} \right]_{(l,m)}. \end{aligned} \tag{6.3c}$$

The matrix $\mathbf{M}_\rho(\underline{\tilde{T}})$ is the same as in (6.1a). Again, the matrices $\mathbf{M}_\rho(\underline{\tilde{T}})$, $\mathbf{A}_{CT}(\underline{\tilde{\rho}})$ and $\mathbf{A}_{DT}(\underline{\tilde{\lambda}})$ depend on some unknowns and need to be updated during the solution procedure.

All affine-linear offsets and source terms are summarized as follows

$$\underline{b}_i := -\underline{b}_{C_i} + \underline{b}_{D1_i} + \underline{b}_{D2_i} + \underline{b}_{D3_i} - \underline{b}_{P_i} - \underline{b}_{f_i}, \tag{6.4a}$$

$$\underline{b}_B := -\sum_{j=1}^{d} \underline{b}_{B_j}, \tag{6.4b}$$

$$\underline{b}_T := -\underline{b}_{CT} + \underline{b}_{DT}. \tag{6.4c}$$

Using the definitions above and dropping the dependencies of the matrices, we can write the semidiscrete system of equations as

$$\frac{\partial}{\partial t} \left(\mathbf{M}_\rho \underline{\tilde{u}}_i \right) + \left(\mathbf{A}_C - \frac{1}{Re} \mathbf{A}_{D1} \right) \underline{\tilde{u}}_i - \frac{1}{Re} \left(\mathbf{A}_{D2_{ij}} - \frac{2}{3} \mathbf{A}_{D3_{ij}} \right) \underline{\tilde{u}}_j + \mathbf{A}_{P_i} \underline{\tilde{p}} = \underline{b}_i, \tag{6.5a}$$

$$\frac{\partial \underline{\tilde{\rho}}}{\partial t} + \mathbf{B}_j \underline{\tilde{u}}_j = \underline{b}_B, \tag{6.5b}$$

$$\frac{1}{\kappa}\frac{\partial}{\partial t}\left(\mathbf{M}_{\rho}\underline{\tilde{T}}\right)+\left(\mathbf{A}_{CT}-\frac{1}{RePr}\mathbf{A}_{DT}\right)\underline{\tilde{T}}=\underline{b}_{T}. \tag{6.5c}$$

6.2.2 Numerical fluxes momentum equation

In this section, we describe the numerical fluxes for the momentum equation, i.e. the convective operator, the diffusive terms and the pressure gradient.

6.2.2.1 Convective operator

Like for incompressible flows we use the local Lax-Friedrichs flux for the convective operator (6.1b), which for variable density flows is given by

$$\begin{aligned}\hat{F}_{ij}^{C}&=\left\{\rho_h u_{hi}u_{hj}\right\}+\frac{1}{2}\Lambda_{K,e}\left[\!\left[u_{hi}\right]\!\right]\\&=\begin{cases}\frac{1}{2}\left(\rho_h^{+}u_{hi}^{+}u_{hj}^{+}+\rho_h^{-}u_{hi}^{-}u_{hj}^{-}\right)+\frac{1}{2}\Lambda_{K,e}\left(u_{hi}^{+}n_j^{+}+u_{hi}^{-}n_j^{-}\right), & \text{on } e\subseteq\Gamma_I\cup\partial\Omega_P,\\ \frac{1}{2}\left(\rho_h^{+}u_{hi}^{+}u_{hj}^{+}+\rho_D u_{Di}u_{Dj}\right)+\frac{1}{2}\Lambda_{K,e}\left(u_{hi}^{+}-u_{Di}\right)n_j^{+}, & \text{on } e\subseteq\partial\Omega_D\cup\partial\Omega_{DN},\\ \rho_h^{+}u_{hi}^{+}u_{hj}^{+}, & \text{on } e\subseteq\partial\Omega_N\cup\Omega_{PO}.\end{cases}\end{aligned} \tag{6.6}$$

Note that the density in (6.6) is evaluated as a function of the temperature, i.e. $\rho_h^{\pm}=p_0/T_h^{\pm}$ and $\rho_D=p_0/T_D$. For multiphase flows the density depends on the level set $\varphi^{\pm}$ and is calculated by (2.6a). In general, it is not guaranteed that during the solution procedure the value for the level set stays within the limits $\varphi_h\in[0,1]$. Therefore, a simple clipping is applied for evaluating the density

$$\rho_h=\begin{cases}\rho_{\mathcal{B}}, & \varphi_h<0,\\ \rho_{\mathcal{A}}\varphi_h+\rho_{\mathcal{B}}(1-\varphi_h), & 0\leq\varphi_h\leq 1,\\ \rho_{\mathcal{A}}, & \varphi_h>1.\end{cases} \tag{6.7}$$

The parameter $\Lambda_{K,e}$ in (6.6) is calculated as

$$\Lambda_{K,e}=\max\left\{|\lambda|;\lambda\in\operatorname{spec}\left(\mathbf{Q}\left(\overline{\rho_h^{+}},\overline{u_{hi}^{+}}\right)\right)\cup\operatorname{spec}\left(\mathbf{Q}\left(\overline{\rho_h^{-}},\overline{u_{hi}^{-}}\right)\right)\right\}, \tag{6.8}$$

where $\overline{\rho_h^{\pm}}$ and $\overline{u_{hi}^{\pm}}$ are the mean values of $\rho_h^{\pm}$ and $u_{hi}^{\pm}$ in $K^{\pm}$, respectively. The flux Jacobian is given by

$$\mathbf{Q}(\rho_h,u_{hi})=\frac{\partial\rho_h u_{hi}u_{hk}n_k}{\partial u_{hj}}, \tag{6.9}$$

leading to

$$\Lambda_{K,e} = \max\left\{2\overline{\rho_h^+}|\overline{u_{hk}^+}n_k^+|, 2\overline{\rho_h^-}|\overline{u_{hk}^-}n_k^-|\right\}. \tag{6.10}$$

6.2.2.2 Diffusive operator

The diffusive operator consists of three terms for low-Mach number flows (6.1c)-(6.1e). All terms are discretized applying the SIP method. The first term also occurs for incompressible flows (cf. Eq. (4.8)), but here we have to consider variable viscosity μ. The numerical flux for the first term (6.1c) reads

$$\begin{aligned}
\hat{F}_i^{D1} &= \left\{\mu\frac{\partial u_{hi}}{\partial x_j}\right\}[\![v^{l,m}]\!] + \left\{\mu\frac{\partial v^{l,m}}{\partial x_j}\right\}[\![u_{hi}]\!] - \mu_{\max}\eta[\![u_{hi}]\!][\![v^{l,m}]\!] \\
&= \begin{cases}
\frac{1}{2}\left(\mu^+\frac{\partial u_{hi}^+}{\partial x_j} + \mu^-\frac{\partial u_{hi}^-}{\partial x_j}\right)\left(\left(v^{l,m}\right)^+ n_j^+ + \left(v^{l,m}\right)^- n_j^-\right) \\
+\frac{1}{2}\left(\mu^+\frac{\partial\left(v^{l,m}\right)^+}{\partial x_j} + \mu^-\frac{\partial\left(v^{l,m}\right)^-}{\partial x_j}\right)\left(u_{hi}^+ n_j^+ + u_{hi}^- n_j^-\right) \\
-\mu_{\max}\eta\left(u_{hi}^+ n_j^+ + u_{hi}^- n_j^-\right)\left(\left(v^{l,m}\right)^+ n_j^+ + \left(v^{l,m}\right)^- n_j^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\
\mu^+\frac{\partial u_{hi}^+}{\partial x_j} n_j^+ \left(v^{l,m}\right)^+ + \mu^+\frac{\partial\left(v^{l,m}\right)^+}{\partial x_j} n_j^+ \left(u_{hi}^+ - u_{Di}\right) \\
-\mu^+\eta\left(u_{hi}^+ - u_{Di}\right)\left(v^{l,m}\right)^+, & \text{on } e \subseteq \partial\Omega_D \cup \Omega_{DN}, \\
0, & \text{on } e \subseteq \partial\Omega_N, \\
\mu^+\frac{\partial u_{hi}^+}{\partial x_j} n_j^+ \left(v^{l,m}\right)^+, & \text{on } e \subseteq \partial\Omega_{PO},
\end{cases}
\end{aligned} \tag{6.11}$$

where $\mu_{\max} = \max\{\mu^+, \mu^-\}$. The penalty parameter η in (6.11) is the same as for incompressible flows given by (4.9)-(4.12). The viscosity $\mu^\pm$ in (6.11) is evaluated as a function of the temperature using Sutherland's law (2.16). For multiphase flows the viscosity is evaluated as a function of the level set $\varphi_h^\pm$ according to (2.6b), where, like for the density, a simple clipping is applied

$$\mu_h = \begin{cases} \mu_{\mathcal{B}}, & \varphi_h < 0, \\ \mu_{\mathcal{A}}\varphi_h + \mu_{\mathcal{B}}(1-\varphi_h), & 0 \le \varphi_h \le 1, \\ \mu_{\mathcal{A}}, & \varphi_h > 1. \end{cases} \tag{6.12}$$

The numerical flux for the second diffusive term (6.1d) is given by

$$\hat{F}_i^{D2} = \left\{ \mu \frac{\partial u_{hj}}{\partial x_i} \right\} [\![v_i^{l,m}]\!] + \left\{ \mu \frac{\partial v_j^{l,m}}{\partial x_i} \right\} [\![u_{hi}]\!] - \mu_{\max} \eta [\![u_{hi}]\!] [\![v_i^{l,m}]\!] \text{ for } i = 1, \ldots, d$$

$$= \begin{cases} \frac{1}{2} \left(\mu^+ \frac{\partial u_{hj}^+}{\partial x_i} + \mu^- \frac{\partial u_{hj}^-}{\partial x_i} \right) \left(\left(v_i^{l,m} \right)^+ n_j^+ + \left(v_i^{l,m} \right)^- n_j^- \right) \\ + \frac{1}{2} \left(\mu^+ \frac{\partial \left(v_j^{l,m} \right)^+}{\partial x_i} + \mu^- \frac{\partial \left(v_j^{l,m} \right)^-}{\partial x_i} \right) \left(u_{hi}^+ n_j^+ + u_{hi}^- n_j^- \right) \\ - \mu_{\max} \eta \left(u_{hi}^+ n_j^+ + u_{hi}^- n_j^- \right) \left(\left(v_i^{l,m} \right)^+ n_j^+ + \left(v_i^{l,m} \right)^- n_j^- \right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ \\ \mu^+ \frac{\partial u_{hj}^+}{\partial x_i} n_j^+ \left(v_i^{l,m} \right)^+ + \mu^+ \frac{\partial \left(v_j^{l,m} \right)^+}{\partial x_i} n_j^+ \left(u_{hi}^+ - u_{Di} \right) \\ - \mu^+ \eta \left(u_{hi}^+ - u_{Di} \right) \left(v_i^{l,m} \right)^+, & \text{on } e \subseteq \partial\Omega_D \cup \Omega_{DN}, \\ \\ \mu^+ \frac{\partial u_{hj}^+}{\partial x_i} n_j^+ \left(v_i^{l,m} \right)^+, & \text{on } e \subseteq \partial\Omega_N \cup \partial\Omega_{PO}. \end{cases} \tag{6.13}$$

As already mentioned above, this term couples the equations for the single velocity components u_{hi}. The symmetry term of $\hat{F}_i^{D2}$, i.e. the second summand, introduces a penalization of jumps for all velocity components in the momentum equation in each direction. In the SIMPLE algorithm we solve the equations in a segregated manner. Therefore, the symmetry term is dropped for $i \neq j$. At an outflow boundary $\partial\Omega_N$ and $\partial\Omega_{PO}$ the flux is evaluated by taking the inner values, i.e. no boundary condition is implied for this operator (cf. the discussion of boundary conditions for incompressible flows in Section 4.4).

The third diffusive term (6.1e) is discretized in the same manner like the first two

$$\hat{F}_i^{D3} = \left\{ \mu \frac{\partial u_{hj}}{\partial x_j} \right\} [\![v_i^{l,m}]\!] + \left\{ \mu \frac{\partial v_j^{l,m}}{\partial x_j} \right\} [\![u_{hi}]\!] - \mu_{\max} \eta [\![u_{hi}]\!] [\![v_i^{l,m}]\!] \text{ for } i = 1, \ldots, d$$

$$
= \begin{cases}
\frac{1}{2}\left(\mu^+\frac{\partial u_{hj}^+}{\partial x_j} + \mu^-\frac{\partial u_{hj}^-}{\partial x_j}\right)\left(\left(v_i^{l,m}\right)^+ n_i^+ + \left(v_i^{l,m}\right)^- n_i^-\right) \\
+\frac{1}{2}\left(\mu^+\frac{\partial\left(v_j^{l,m}\right)^+}{\partial x_j} + \mu^-\frac{\partial\left(v_j^{l,m}\right)^-}{\partial x_j}\right)\left(u_{hi}^+ n_i^+ + u_{hi}^- n_i^-\right) \\
-\mu_{\max}\eta\left(u_{hi}^+ n_j^+ + u_{hi}^- n_j^-\right)\left(\left(v_i^{l,m}\right)^+ n_j^+ + \left(v_i^{l,m}\right)^- n_j^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\
\mu^+\frac{\partial u_{hj}^+}{\partial x_j} n_i^+ \left(v_i^{l,m}\right)^+ + \mu^+\frac{\partial\left(v_j^{l,m}\right)^+}{\partial x_j} n_i^+ \left(u_{hi}^+ - u_{Di}\right) \\
-\mu^+\eta\left(u_{hi}^+ - u_{Di}\right)\left(v_i^{l,m}\right)^+, & \text{on } e \subseteq \partial\Omega_D \cup \Omega_{DN}, \\
\mu^+\frac{\partial u_{hj}^+}{\partial x_j} n_i^+ \left(v_i^{l,m}\right)^+, & \text{on } e \subseteq \partial\Omega_N \cup \partial\Omega_{PO}.
\end{cases}
\tag{6.14}
$$

The symmetry term and the boundary conditions of $\hat{F}_i^{D3}$ are treated in the same way like discussed above for the numerical flux $\hat{F}_i^{D2}$. Note that $\hat{F}_i^{D3}$ only occurs for low-Mach number flows and is zero for multiphase flows.

6.2.2.3 Gradient operator

The pressure gradient operator (6.1f) is like in the incompressible case discretized by a central difference flux, which is given by

$$
\begin{aligned}
\hat{F}^P &= \{p_h\} \\
&= \begin{cases}
\frac{1}{2}\left(p_h^+ + p_h^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\
p_h^+, & \text{on } e \subseteq \partial\Omega_D \cup \partial\Omega_{DN}, \\
0, & \text{on } e \subseteq \partial\Omega_N, \\
p_D, & \text{on } e \subseteq \partial\Omega_{PO}.
\end{cases}
\end{aligned}
\tag{6.15}
$$

6.2.3 Numerical fluxes continuity equation

The divergence operator (6.2b) is like the pressure gradient discretized by a central difference flux

$$
\hat{F}_j^{\mathrm{Div}} = \{\rho_h u_{hj}\}
$$

$$= \begin{cases} \frac{1}{2}\left(\rho_h^+ u_{hj}^+ + \rho_h^- u_{hj}^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ \rho_D u_{Dj}, & \text{on } e \subseteq \partial\Omega_D \cup \partial\Omega_{DN}, \\ \rho^+ u_{hj}^+, & \text{on } e \subseteq \partial\Omega_N \cup \Omega_{PO}. \end{cases} \tag{6.16}$$

The density in (6.16) is evaluated as a function of the temperature like discussed above. For multiphase flows the continuity equation is the same like for incompressible flows with the numerical flux given by (4.14).

6.2.4 Numerical fluxes scalar equation

In this section the numerical fluxes for the scalar equation is given. For low-Mach number flows we have to consider advection as well as diffusion and for multiphase flows there is only advection.

6.2.4.1 Advective operator

The advection operator (6.3b) is discretized like the convection in the momentum equation using the local Lax-Friedrichs flux

$$\hat{F}_j^{CT} = \{\rho_h u_{hj} T_h\} + \frac{1}{2}\Lambda_{K,e}[\![T_h]\!]$$

$$= \begin{cases} \frac{1}{2}\left(\rho_h^+ u_{hj}^+ T_h^+ + \rho_h^- u_{hj}^- T_h^-\right) + \frac{1}{2}\Lambda_{K,e}\left(T_h^+ n_j^+ + T_h^- n_j^-\right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\ \frac{1}{2}\left(\rho_h^+ u_{hj}^+ T_h^+ + \rho_D u_{Dj} T_D\right) + \frac{1}{2}\Lambda_{K,e}\left(T_h^+ - T_D\right) n_j^+, & \text{on } e \subseteq \partial\Omega_D, \\ \rho_h^+ u_{hj}^+ T_h^+, & \text{on } e \subseteq \partial\Omega_N \cup \Omega_{DN} \cup \Omega_{PO}, \end{cases} \tag{6.17}$$

The parameter $\Lambda_{K,e}$ is given by

$$\Lambda_{K,e} = \max\left\{|\lambda|; \lambda \in \text{spec}\left(\mathbf{Q}\left(\overline{\rho_h^+}, \overline{u_{hi}^+}, \overline{T_h^+}\right)\right) \cup \text{spec}\left(\mathbf{Q}\left(\overline{\rho_h^-}, \overline{u_{hi}^-}, \overline{T_h^-}\right)\right)\right\}, \tag{6.18}$$

which is evaluated using the mean values in $K^{\pm}$. For low-Mach number flows the flux Jacobian is

$$\mathbf{Q}(\rho_h, u_{hi}, T_h) = \frac{\partial \rho_h u_{hj} T_h n_j}{\partial T_h}, \tag{6.19}$$

which leads to

$$\Lambda_{K,e} = \max\left\{\overline{\rho_h^+}|\overline{u_{hj}^+} n_j^+|, \overline{\rho_h^-}|\overline{u_{hj}^-} n_j^-|\right\}. \tag{6.20}$$

For multiphase flows the advection operator does not contain the density, cf. equation (2.8c). Therefore, the flux Jacobian is

$$\mathbf{Q}(u_{hi}, \varphi_h) = \frac{\partial u_{hj}\varphi_h n_j}{\partial \varphi_h}, \tag{6.21}$$

and the parameter $\Lambda_{K,e}$ is calculated to

$$\Lambda_{K,e} = \max\left\{ |\overline{u_{hj}^+} n_j^+|, |\overline{u_{hj}^-} n_j^-| \right\}. \tag{6.22}$$

6.2.4.2 Diffusive operator

The diffusive operator of the temperature equation is discretized applying the SIP method

$$\begin{aligned}
\hat{F}^{DT} &= \left\{ \lambda \frac{\partial T_h}{\partial x_j} \right\} [\![v^{l,m}]\!] + \left\{ \lambda \frac{\partial v^{l,m}}{\partial x_j} \right\} [\![T_h]\!] - \lambda_{\max} \eta [\![T_h]\!] [\![v^{l,m}]\!] \\
&= \begin{cases}
\frac{1}{2}\left(\lambda^+ \frac{\partial T_h^+}{\partial x_j} + \lambda^- \frac{\partial T_h^-}{\partial x_j} \right) \left(\left(v^{l,m}\right)^+ n_j^+ + \left(v^{l,m}\right)^- n_j^- \right) \\
+ \frac{1}{2}\left(\lambda^+ \frac{\partial \left(v^{l,m}\right)^+}{\partial x_j} + \lambda^- \frac{\partial \left(v^{l,m}\right)^-}{\partial x_j} \right) \left(T_h^+ n_j^+ + T_h^- n_j^- \right) \\
- \lambda_{\max} \eta \left(T_h^+ n_j^+ + T_h^- n_j^- \right) \left(\left(v^{l,m}\right)^+ n_j^+ + \left(v^{l,m}\right)^- n_j^- \right), & \text{on } e \subseteq \Gamma_I \cup \partial\Omega_P, \\
\lambda^+ \frac{\partial T_h^+}{\partial x_j} n_j^+ \left(v^{l,m}\right)^+ + \lambda^+ \frac{\partial \left(v^{l,m}\right)^+}{\partial x_j} n_j^+ \left(T_h^+ - T_D\right) \\
- \lambda^+ \eta \left(T_h^+ - T_D\right) \left(v^{l,m}\right)^+, & \text{on } e \subseteq \partial\Omega_D, \\
0, & \text{on } e \subseteq \partial\Omega_{DN} \cup \partial\Omega_N \cup \partial\Omega_{PO},
\end{cases}
\end{aligned} \tag{6.23}$$

where $\lambda_{\max} = \max\{\lambda^+, \lambda^-\}$ and the penalty parameter η is given by (4.9)-(4.12).

6.3 Temporal discretization

The semidiscrete system of equations (6.5a)-(6.5c) is discretized in time using a BDF scheme. The time interval $[0, T]$ is divided into uniform time steps Δt

$$\begin{aligned}
\frac{\beta_0}{\gamma \Delta t} \mathbf{M}_\rho^{n+1} \underline{\tilde{u}}_i^{n+1} + \frac{1}{\gamma \Delta t} \sum_{\alpha=1}^{s} \beta_\alpha \mathbf{M}_\rho^{n+1-\alpha} \underline{\tilde{u}}_i^{n+1-\alpha} + \left(\mathbf{A}_C^{n+1} - \frac{1}{Re} \mathbf{A}_{D1}^{n+1} \right) \underline{\tilde{u}}_i^{n+1} \\
- \frac{1}{Re} \left(\mathbf{A}_{D2_{ij}}^{n+1} - \frac{2}{3} \mathbf{A}_{D3_{ij}}^{n+1} \right) \underline{\tilde{u}}_j^{n+1} + \mathbf{A}_{P_i} \underline{\tilde{p}}^{n+1} = \underline{b}_i^{n+1},
\end{aligned} \tag{6.24a}$$

$$\frac{1}{\gamma\Delta t}\sum_{\alpha=0}^{s}\beta_\alpha\underline{\tilde{\rho}}^{n+1-\alpha}+\mathbf{B}_j^{n+1}\underline{\tilde{u}}_j^{n+1}=\underline{b}_B^{n+1}, \tag{6.24b}$$

$$\frac{1}{\kappa}\frac{\beta_0}{\gamma\Delta t}\mathbf{M}_\rho^{n+1}\underline{\tilde{T}}^{n+1}+\frac{1}{\kappa}\frac{1}{\gamma\Delta t}\sum_{\alpha=1}^{s}\beta_\alpha\mathbf{M}_\rho^{n+1-\alpha}\underline{\tilde{T}}^{n+1-\alpha} + \left(\mathbf{A}_{CT}^{n+1}-\frac{1}{RePr}\mathbf{A}_{DT}^{n+1}\right)\underline{\tilde{T}}^{n+1}=\underline{b}_T^{n+1}. \tag{6.24c}$$

In (6.24a)-(6.24c) the superscripts $n+1$ and $n+1-\alpha$ denote the current and previous time steps, respectively. The coefficients β_0, β_α and γ for the BDF schemes have already been given in Table 4.1. In the following we introduce some notation to write the discrete system of equations in a more compact form, which will then be used to derive the SIMPLE algorithm. The temporal terms for the previous time steps of the momentum equations (6.24a) and temperature equation (6.24c) are summarized as follows

$$\underline{\sigma}_{u_i}^n:=\sum_{\alpha=1}^{s}\beta_\alpha\mathbf{M}_\rho^{n+1-\alpha}\underline{\tilde{u}}_i^{n+1-\alpha}, \tag{6.25a}$$

$$\underline{\sigma}_T^n:=\sum_{\alpha=1}^{s}\beta_\alpha\mathbf{M}_\rho^{n+1-\alpha}\underline{\tilde{T}}^{n+1-\alpha}. \tag{6.25b}$$

The temporal terms of the continuity equation (6.24b) are summarized in

$$\underline{\sigma}_\rho^{n+1}:=\sum_{\alpha=0}^{s}\beta_\alpha\underline{\tilde{\rho}}^{n+1-\alpha}, \tag{6.26}$$

which also includes the current time step. Within the SIMPLE algorithm this term will be linearized. Also note that this term only occurs for low-Mach number flows. Without loss of generality, the SIMPLE algorithm below will be derived for spatial dimension $d=2$. Therefore, we introduce the following abbreviations for the matrices of the momentum equations (6.24a)

$$\mathbf{A}_{11}:=\frac{\beta_0}{\gamma\Delta t}\mathbf{M}_\rho^{n+1}+\mathbf{A}_C-\frac{1}{Re}\left(\mathbf{A}_{D1}+\mathbf{A}_{D2_{11}}-\frac{2}{3}\mathbf{A}_{D3_{11}}\right), \tag{6.27a}$$

$$\mathbf{A}_{12}:=-\frac{1}{Re}\left(\mathbf{A}_{D2_{12}}-\frac{2}{3}\mathbf{A}_{D3_{12}}\right), \tag{6.27b}$$

$$\mathbf{A}_{22}:=\frac{\beta_0}{\gamma\Delta t}\mathbf{M}_\rho^{n+1}+\mathbf{A}_C-\frac{1}{Re}\left(\mathbf{A}_{D1}+\mathbf{A}_{D2_{22}}-\frac{2}{3}\mathbf{A}_{D3_{22}}\right), \tag{6.27c}$$

$$\mathbf{A}_{21}:=-\frac{1}{Re}\left(\mathbf{A}_{D2_{21}}-\frac{2}{3}\mathbf{A}_{D3_{21}}\right), \tag{6.27d}$$

and in the same manner for the temperature equation

$$\mathbf{C}_T := \frac{1}{\kappa}\frac{\beta_0}{\gamma\Delta t}\mathbf{M}_\rho^{n+1} + \mathbf{A}_{CT} - \frac{1}{RePr}\mathbf{A}_{DT}. \tag{6.28}$$

Then, we can write the discrete system of equations in the final form

$$\begin{pmatrix} \mathbf{A}_{11}^{n+1} & \mathbf{A}_{12}^{n+1} & \mathbf{A}_{P_1} & 0 \\ \mathbf{A}_{21}^{n+1} & \mathbf{A}_{22}^{n+1} & \mathbf{A}_{P_2} & 0 \\ \mathbf{B}_1^{n+1} & \mathbf{B}_2^{n+1} & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_T^{n+1} \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^{n+1} \\ \underline{\tilde{u}}_2^{n+1} \\ \underline{\tilde{p}}^{n+1} \\ \underline{\tilde{T}}^{n+1} \end{pmatrix} = \begin{pmatrix} \underline{b}_1^{n+1} - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n \\ \underline{b}_2^{n+1} - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n \\ \underline{b}_B^{n+1} - \frac{1}{\gamma\Delta t}\underline{\sigma}_\rho^{n+1} \\ \underline{b}_T^{n+1} - \frac{1}{\gamma\Delta t}\underline{\sigma}_T^n \end{pmatrix}. \tag{6.29}$$

The SIMPLE algorithm for solving (6.29) will be discussed in Section (6.5).

6.4 Jump conditions multiphase flows

For two-phase flows one can derive the following jump conditions at the interface $\mathcal{I}$ of the two phases $\mathcal{A}$ and $\mathcal{B}$ (cf. (Wang & Oberlack 2011))

$$[\![u_i]\!]_{\mathcal{I}} = 0, \tag{6.30a}$$

$$\left[\!\!\left[p\delta_{ij} - \frac{\mu}{Re}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)\right]\!\!\right]_{\mathcal{I}} n_j = 0, \tag{6.30b}$$

where n_j shall denote the normal vector at the interface. Since no surface tension is applied in this work, the right-hand side of (6.30b) is zero. The jump operator at the interface is defined as $[\![c]\!]_{\mathcal{I}} := c_{\mathcal{A}} - c_{\mathcal{B}}$.

The aim of this section is to show that the numerical fluxes are consistent with the jump conditions at the interface for two-phase flows. The procedure is the same like in Section 4.4, where we discussed the boundary conditions for incompressible flows. Like in Section 4.4, we refer to the inner and outer cell values at some edge $e \in \partial K^l$ by the superscripts "+" and "-", respectively, which implies for the normal vectors $n_{ej} = n^+ = -n^-$. For the derivation we use vector-valued test functions $v_i^{l,m} \in (V_k)^d$. Again, we drop the superscript indicating inner and outer values for the test function, since the test functions are only non-zero inside the associated cell K^l.

The jump conditions (6.30a) and (6.30b) involve the viscous terms and the pressure. Therefore, we start from the Stokes equations in the following form

$$-\frac{1}{Re}\frac{\partial}{\partial x_j}\left(\mu\frac{\partial u_i}{\partial x_j} + \mu\frac{\partial u_j}{\partial x_i}\right) + \frac{\partial p}{\partial x_i} = 0, \tag{6.31}$$

which reads in the weak form

$$\int_{\partial K^l} -\frac{1}{Re}\left(\hat{F}_i^{D1} + \hat{F}_i^{D2}\right) + \hat{F}^P n_{ei} v_i^{l,m}\, ds$$

$$- \int_{K^l} -\frac{1}{Re} \left(\mu \frac{\partial u_{hi}}{\partial x_j} \frac{\partial v_i^{l,m}}{\partial x_j} + \mu \frac{\partial u_{hj}}{\partial x_i} \frac{\partial v_i^{l,m}}{\partial x_j} \right) + p_h \frac{\partial v_i^{l,m}}{\partial x_i} \, d\mathbf{x} = 0. \tag{6.32}$$

Next, we derive the strong form by carrying out integration by parts once again for the volume integral in (6.32). The edge integrals of this second integration by parts are evaluated by taking the inner values leading to

$$\begin{aligned}
&\int_{\partial K^l} -\frac{1}{Re} \left(\hat{F}_i^{D1} - \mu^+ \frac{\partial u_{hi}^+}{\partial x_j} v_i^{l,m} n_{ej} + \hat{F}_i^{D2} - \mu^+ \frac{\partial u_{hj}^+}{\partial x_i} v_i^{l,m} n_{ej} \right) ds \\
&+ \int_{\partial K^l} \left(\hat{F}^P - p_h^+ \right) n_{ei} v_i^{l,m} \, ds \\
&+ \int_{K^l} \left(-\frac{1}{Re} \frac{\partial}{\partial x_j} \left(\mu \frac{\partial u_{hi}}{\partial x_j} + \mu \frac{\partial u_{hj}}{\partial x_i} \right) + \frac{\partial p_h}{\partial x_i} \right) v_i^{l,m} \, d\mathbf{x} = 0.
\end{aligned} \tag{6.33}$$

Since for an approximate solution u_{hi} and p_h of (6.31) the volume integral in (6.33) shall vanish, the edge integrals in (6.33) have also to be zero. Inserting the numerical fluxes (6.11), (6.13) and (6.15) in (6.33) and evaluating the edge integrals at the interface $\mathcal{I}$ yields

$$\begin{aligned}
&\int_{\mathcal{I}} -\frac{1}{Re} \left(\frac{1}{2} \left(\mu^- \frac{\partial u_{hi}^-}{\partial x_j} - \mu^+ \frac{\partial u_{hi}^+}{\partial x_j} \right) v_i^{l,m} n_{ej} \right. \\
&\qquad \left. + \left(\frac{1}{2} \mu^+ \frac{\partial v_i^{l,m}}{\partial x_j} n_{ej} - \mu_{\max} \eta v_i^{l,m} \right) \left(u_{hi}^+ - u_{hi}^- \right) \right) ds \\
&+ \int_{\mathcal{I}} -\frac{1}{Re} \left(\frac{1}{2} \left(\mu^- \frac{\partial u_{hj}^-}{\partial x_i} - \mu^+ \frac{\partial u_{hj}^+}{\partial x_i} \right) v_i^{l,m} n_{ej} \right. \\
&\qquad \left. + \left(\frac{1}{2} \mu^+ \frac{\partial v_j^{l,m}}{\partial x_i} n_{ej} - \mu_{\max} \eta v_i^{l,m} \right) \left(u_{hi}^+ - u_{hi}^- \right) \right) ds \\
&+ \int_{\mathcal{I}} \frac{1}{2} \left(p^- - p^+ \right) n_{ei} v_i^{l,m} \, ds \stackrel{!}{=} 0,
\end{aligned} \tag{6.34}$$

which in turn can be rewritten as

$$\begin{aligned}
&- \frac{1}{2} \int_{\mathcal{I}} \left[\!\!\left[p_h \delta_{ij} - \frac{\mu}{Re} \left(\frac{\partial u_{hi}}{\partial x_j} + \frac{\partial u_{hj}}{\partial x_i} \right) \right]\!\!\right]_{\mathcal{I}} n_{ej} v_i^{l,m} \, ds \\
&- \frac{1}{Re} \int_{\mathcal{I}} \left(\frac{1}{2} \mu^+ \left(\frac{\partial v_i^{l,m}}{\partial x_j} + \frac{\partial v_j^{l,m}}{\partial x_i} \right) n_{ej} - 2 \mu_{\max} \eta v_i^{l,m} \right) [\![u_{hi}]\!]_{\mathcal{I}} \, ds \stackrel{!}{=} 0.
\end{aligned} \tag{6.35}$$

It is obvious that (6.35) is fulfilled for the jump conditions (6.30a)-(6.30b), showing that the numerical fluxes are consistent with the jump conditions. Like discussed in Section 2.2, we consider a smooth interface approach in this work, i.e. the numerical fluxes reproduce the correct jump conditions in the limit of zero interface thickness ϵ.

In the context of a sharp interface model the jump conditions are exactly represented by the numerical flux at the interface, cf. (Kummer 2013) and (Heimann et al. 2013). In Section 7.1.2, we will show numerical results where the interface is aligned exactly with the grid edges. By doing that, we can realize a sharp interface demonstrating that the jump conditions are fulfilled exactly.

Note that for a sharp interface model the equation (6.31) in the bulk can be reduced to

$$-\frac{\mu}{Re}\frac{\partial^2 u_i}{\partial x_j \partial x_j} + \frac{\partial p}{\partial x_i} = 0. \tag{6.36}$$

If using (6.36) instead of (6.31) the numerical flux for the velocity gradient has to be modified to account also for the jump of the transposed velocity gradient at the interface, which leads to a non-symmetric operator. For details see (Kummer 2013) and (Heimann et al. 2013).

6.5 SIMPLE algorithm

In this section, we derive the SIMPLE algorithm for solving the discrete system of equations (6.29) for the primitive variables $\underline{\tilde{u}}_1^{n+1}$, $\underline{\tilde{u}}_2^{n+1}$, $\underline{\tilde{p}}^{n+1}$ and $\underline{\tilde{T}}^{n+1}$ at the new time step. The procedure is the same like for incompressible flows in Section 4.5. We introduce the following iterative process

$$\begin{aligned}\{\underline{\tilde{u}}(t_n), \underline{\tilde{p}}(t_n), \underline{\tilde{T}}(t_n)\} =: \{\underline{\tilde{u}}_i^0, \underline{\tilde{p}}^0, \underline{\tilde{T}}^0\} \mapsto \ldots \mapsto \{\underline{\tilde{u}}_i^\vartheta, \underline{\tilde{p}}^\vartheta, \underline{\tilde{T}}^\vartheta\} \mapsto \\ \mapsto \{\underline{\tilde{u}}_i^{\vartheta+1}, \underline{\tilde{p}}^{\vartheta+1}, \underline{\tilde{T}}^{\vartheta+1}\} \mapsto \ldots \xrightarrow{\lim} \{\underline{\tilde{u}}(t_{n+1}), \underline{\tilde{p}}(t_{n+1}), \underline{\tilde{T}}(t_{n+1})\},\end{aligned}$$

where the inner SIMPLE iterations in each time step are represented by the superscript ϑ. For $\vartheta = 0$ the values from the previous time step t_n are taken. The unknowns at iteration step $\vartheta + 1$ are determined by

$$\begin{pmatrix} \mathbf{A}_{11}^\vartheta & 0 & \mathbf{A}_{P_1} & 0 \\ 0 & \mathbf{A}_{22}^\vartheta & \mathbf{A}_{P_2} & 0 \\ \mathbf{B}_1^\vartheta & \mathbf{B}_2^\vartheta & 0 & 0 \\ 0 & 0 & 0 & \mathbf{C}_T^{\vartheta+\frac{1}{2}} \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^{\vartheta+1} \\ \underline{\tilde{u}}_2^{\vartheta+1} \\ \underline{\tilde{p}}^{\vartheta+1} \\ \underline{\tilde{T}}^{\vartheta+1} \end{pmatrix} = \begin{pmatrix} \underline{b}_1^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n - \mathbf{A}_{12}^\vartheta \underline{\tilde{u}}_2^\vartheta \\ \underline{b}_2^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n - \mathbf{A}_{21}^\vartheta \underline{\tilde{u}}_1^\vartheta \\ \underline{b}_B^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_\rho^\vartheta \\ \underline{b}_T^{\vartheta+\frac{1}{2}} - \frac{1}{\gamma\Delta t}\underline{\sigma}_T^n \end{pmatrix}. \tag{6.37}$$

All matrices and affine-linear offsets in (6.37) are evaluated using the values from the previous iteration step, i.e. the system of equations (6.37) is linear. The momentum equations for the velocity components $\underline{\tilde{u}}_1^{\vartheta+1}$ and $\underline{\tilde{u}}_2^{\vartheta+1}$ are decoupled by evaluating the off-diagonal terms $\mathbf{A}_{12}^\vartheta \underline{\tilde{u}}_2^\vartheta$ and $\mathbf{A}_{21}^\vartheta \underline{\tilde{u}}_1^\vartheta$ at the previous iteration step. The temporal term in the continuity equation is calculated by

$$\underline{\sigma}_\rho^\vartheta = \beta_0 \underline{\tilde{\rho}}^\vartheta + \sum_{\alpha=1}^{s} \beta_\alpha \underline{\tilde{\rho}}^{n+1-\alpha}. \tag{6.38}$$

We first solve the momentum and continuity equation by the SIMPLE algorithm. Then, the new velocity is used to update the convective operator of the temperature equation, i.e.

$$\mathbf{C}_T^{\vartheta+\frac{1}{2}} = \frac{1}{\kappa}\frac{\beta_0}{\gamma\Delta t}\mathbf{M}_\rho^\vartheta + \mathbf{A}_{CT}(\underline{\tilde{\rho}}^\vartheta \underline{\tilde{u}}^{\vartheta+1}) - \frac{1}{RePr}\mathbf{A}_{DT}(\underline{\tilde{\lambda}}^\vartheta). \tag{6.39}$$

Then in the next step, the temperature equation is solved for the new temperature.

Within the SIMPLE algorithm we split the velocity and the pressure in intermediate and correction components

$$\underline{\tilde{u}}_i^{\vartheta+1} = \underline{\tilde{u}}_i^* + \underline{\tilde{u}}_i', \tag{6.40a}$$
$$\underline{\tilde{p}}^{\vartheta+1} = \underline{\tilde{p}}^* + \underline{\tilde{p}}' := \underline{\tilde{p}}^\vartheta + \underline{\tilde{p}}', \tag{6.40b}$$

where for the intermediate pressure $\underline{\tilde{p}}^*$ the value from the previous iteration step $\underline{\tilde{p}}^\vartheta$ is taken. Then, the momentum equations can be solved sequentially for the intermediate velocity components

$$\begin{pmatrix} \mathbf{A}_{11}^\vartheta & 0 \\ 0 & \mathbf{A}_{22}^\vartheta \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1^* \\ \underline{\tilde{u}}_2^* \end{pmatrix} = \begin{pmatrix} \underline{b}_1^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n - \mathbf{A}_{12}^\vartheta \underline{\tilde{u}}_2^\vartheta \\ \underline{b}_2^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n - \mathbf{A}_{21}^\vartheta \underline{\tilde{u}}_1^\vartheta \end{pmatrix} - \begin{pmatrix} \mathbf{A}_{P_1} \\ \mathbf{A}_{P_2} \end{pmatrix} \underline{\tilde{p}}^\vartheta. \tag{6.41}$$

By subtracting (6.41) from the momentum equations of (6.37) and inserting the decomposition for velocity (6.40a) into the continuity equation of (6.37), we obtain the linear system for the correction components

$$\begin{pmatrix} \mathbf{A}_{11}^\vartheta & 0 & \mathbf{A}_{P_1} \\ 0 & \mathbf{A}_{22}^\vartheta & \mathbf{A}_{P_2} \\ \mathbf{B}_1^\vartheta & \mathbf{B}_2^\vartheta & 0 \end{pmatrix} \begin{pmatrix} \underline{\tilde{u}}_1' \\ \underline{\tilde{u}}_2' \\ \underline{\tilde{p}}' \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ \underline{b}_B^\vartheta - \frac{1}{\gamma\Delta t}\underline{\sigma}_\rho^\vartheta - \mathbf{B}_1^\vartheta \underline{\tilde{u}}_1^* - \mathbf{B}_2^\vartheta \underline{\tilde{u}}_2^* \end{pmatrix}. \tag{6.42}$$

Then, the diagonal parts in the momentum equations are approximated by

$$\mathcal{A}_{11}^\vartheta \approx \mathbf{A}_{11}^\vartheta, \tag{6.43a}$$
$$\mathcal{A}_{22}^\vartheta \approx \mathbf{A}_{22}^\vartheta, \tag{6.43b}$$

leading to the following expression for the correction components of the velocity

$$\begin{pmatrix} \underline{\tilde{u}}_1' \\ \underline{\tilde{u}}_2' \end{pmatrix} = -\begin{pmatrix} \left(\mathcal{A}_{11}^\vartheta\right)^{-1} & 0 \\ 0 & \left(\mathcal{A}_{22}^\vartheta\right)^{-1} \end{pmatrix} \begin{pmatrix} \mathbf{A}_{P_1} \\ \mathbf{A}_{P_2} \end{pmatrix} \underline{\tilde{p}}'. \tag{6.44}$$

Inserting (6.44) in the continuity equation of (6.42) yields a linear equation for the pressure correction

$$\left(\mathbf{B}_1^\vartheta \left(\mathcal{A}_{11}^\vartheta\right)^{-1} \mathbf{A}_{P_1} + \mathbf{B}_2^\vartheta \left(\mathcal{A}_{22}^\vartheta\right)^{-1} \mathbf{A}_{P_2}\right) \underline{\tilde{p}}' = -\underline{b}_B^\vartheta + \frac{1}{\gamma\Delta t}\underline{\sigma}_\rho^\vartheta + \mathbf{B}_1^\vartheta \underline{\tilde{u}}_1^* + \mathbf{B}_2^\vartheta \underline{\tilde{u}}_2^*. \tag{6.45}$$

Having solved (6.45) for the pressure correction, the velocity correction is calculated by (6.44) and the updated velocity and pressure are obtained by (6.40a) and (6.40b). Like already mentioned above the new velocity is used to update the matrix of the temperature equation $\mathbf{C}_T^{\vartheta+\frac{1}{2}}$. Then, in the last step of the algorithm the new temperature is obtained by solving

$$\mathbf{C}_T^{\vartheta+\frac{1}{2}}\underline{\tilde{T}}^{\vartheta+1} = \underline{b}_T^{\vartheta+\frac{1}{2}} - \frac{1}{\gamma\Delta t}\underline{\sigma}_T^n. \tag{6.46}$$

For the approximations in the corrector step (6.43a) and (6.43b) we have considered two choices. The first one is taking the diagonal matrix

$$\left(\mathcal{A}_{\alpha\alpha}^{\vartheta}\right)_{i,j} = \begin{cases} \left(\mathbf{A}_{\alpha\alpha}^{\vartheta}\right)_{i,j}, & \text{if } i = j, \\ 0, & \text{otherwise.} \end{cases} \tag{6.47}$$

In (6.47) $i \in [1, N_k N]$ and $j \in [1, N_k N]$ are the row and cell indices, respectively. Note that there is no summation over α in (6.47). The second option for the approximation is taking a block diagonal matrix

$$\left(\mathcal{A}_{\alpha\alpha}^{\vartheta}\right)_{i,j} = \begin{cases} \left(\mathbf{A}_{\alpha\alpha}^{\vartheta}\right)_{i,j}, & \text{if floor}\left(\frac{i-1}{N_k}\right) N_k < j \leq \text{floor}\left(\frac{i-1}{N_k}\right) N_k + N_k, \\ 0, & \text{otherwise,} \end{cases} \tag{6.48}$$

which can be understood to be the direct counterpart of the FVM version of the SIMPLE algorithm, where in the correction step the coupling to all neighboring cells is neglected. In (6.44) and (6.45) the inverse of these approximations is needed. For the first option, taking a diagonal matrix this inversion is trivial. For the second option, the diagonal blocks, which are of dimension $N_k \times N_k$, need to be inverted, which is also feasible since N_k is rather small.

6.5.1 Under-relaxation

Like in the incompressible case, we need under-relaxation to stabilize the algorithm. Therefore, the equations for the intermediate velocity (6.41) are replaced using implicit under-relaxation

$$\left[\left(\frac{1-\alpha_u}{\alpha_u}\right)\mathcal{A}_{11}^{\vartheta} + \mathbf{A}_{11}^{\vartheta}\right]\underline{\tilde{u}}_1^{*} = \underline{b}_1^{\vartheta} - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_1}^n - \mathbf{A}_{12}^{\vartheta}\underline{\tilde{u}}_2^{\vartheta} - \mathbf{A}_{P_1}\underline{\tilde{p}} + \left(\frac{1-\alpha_u}{\alpha_u}\right)\mathcal{A}_{11}^{\vartheta}\underline{\tilde{u}}_1^{\vartheta}, \tag{6.49a}$$

$$\left[\left(\frac{1-\alpha_u}{\alpha_u}\right)\mathcal{A}_{22}^{\vartheta} + \mathbf{A}_{22}^{\vartheta}\right]\underline{\tilde{u}}_2^{*} = \underline{b}_2^{\vartheta} - \frac{1}{\gamma\Delta t}\underline{\sigma}_{u_2}^n - \mathbf{A}_{21}^{\vartheta}\underline{\tilde{u}}_1^{\vartheta} - \mathbf{A}_{P_2}\underline{\tilde{p}} + \left(\frac{1-\alpha_u}{\alpha_u}\right)\mathcal{A}_{22}^{\vartheta}\underline{\tilde{u}}_2^{\vartheta}, \tag{6.49b}$$

where $\alpha_u \in (0,1]$ is the under-relaxation factor for the velocity. The under-relaxation for the pressure is done explicitly by

$$\underline{\tilde{p}}^{\vartheta+1} = \underline{\tilde{p}}^{\vartheta} + \alpha_p \underline{\tilde{p}}', \tag{6.50}$$

with $\alpha_p \in (0,1]$ being the under-relaxation factor for the pressure. The under-relaxation for the temperature is also done explicitly. Therefore, the equation for the temperature (6.46) is solved for some intermediate value $\underline{\tilde{T}}^*$

$$\mathbf{C}_T^{\vartheta+\frac{1}{2}} \underline{\tilde{T}}^* = \underline{b}_T^{\vartheta+\frac{1}{2}} - \frac{1}{\gamma \Delta t} \underline{\sigma}_T^n. \tag{6.51}$$

Then, the new temperature is calculated by

$$\underline{\tilde{T}}^{\vartheta+1} = (1-\alpha_T)\underline{\tilde{T}}^{\vartheta} + \alpha_T \underline{\tilde{T}}^*, \tag{6.52}$$

where $\alpha_T \in (0,1]$ is the under-relaxation factor for the temperature.

6.5.2 Summary of the SIMPLE algorithm

The discrete system of equations (6.29) is solved by the iterative procedure

$$\begin{aligned}\{\underline{\tilde{u}}(t_n), \underline{\tilde{p}}(t_n), \underline{\tilde{T}}(t_n)\} =: \{\underline{\tilde{u}}_i^0, \underline{\tilde{p}}^0, \underline{\tilde{T}}^0\} \mapsto \ldots \mapsto \{\underline{\tilde{u}}_i^{\vartheta}, \underline{\tilde{p}}^{\vartheta}, \underline{\tilde{T}}^{\vartheta}\} \mapsto \\ \mapsto \{\underline{\tilde{u}}_i^{\vartheta+1}, \underline{\tilde{p}}^{\vartheta+1}, \underline{\tilde{T}}^{\vartheta+1}\} \mapsto \ldots \xrightarrow{\lim} \{\underline{\tilde{u}}(t_{n+1}), \underline{\tilde{p}}(t_{n+1}), \underline{\tilde{T}}(t_{n+1})\},\end{aligned}$$

until all convergence criteria are fulfilled

$$\begin{aligned}&\|p_h'\|_{L^2(\Omega)} \leq \epsilon_p, \\ &\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} \leq \epsilon_u, \\ &\|T_h^{\vartheta+1} - T_h^{\vartheta}\|_{L^2(\Omega)} \leq \epsilon_T,\end{aligned}$$

where ϵ_p, ϵ_u and ϵ_T are predefined tolerances. The steps for one SIMPLE iteration within one time step are:

1. Solve the momentum equations (6.49a) and (6.49b) sequentially for the intermediate velocity components $\underline{\tilde{u}}_i^*$.
2. Solve (6.45) for the pressure correction $\underline{\tilde{p}}'$.
3. Update the velocity $\underline{\tilde{u}}_i^{\vartheta+1}$ via (6.40a) using (6.44) and the pressure $\underline{\tilde{p}}^{\vartheta+1}$ via (6.50).
4. Solve the temperature equation (6.51) and update the temperature $\underline{\tilde{T}}^{\vartheta+1}$ via (6.52).
5. Update the thermodynamic pressure p_0 via (2.21) - only for confined systems.
6. Update the density $\underline{\tilde{\rho}}^{\vartheta+1}$ via (2.15), the viscosity $\underline{\tilde{\mu}}^{\vartheta+1}$ via (2.16) and the thermal conductivity $\underline{\tilde{\lambda}}^{\vartheta+1}$ via (2.17).

7 Numerical results for variable density flows

In this chapter, we present the numerical results for multiphase flows, cf. Section 7.1, and low-Mach number flows, cf. Section 7.2. The temporal and spatial discretization as well as the algorithm to solve the non-linear discrete system of equations have been described in Chapter 6. Like in the incompressible case, all linear systems arising in the SIMPLE algorithm are solved using the sparse direct solver PARDISO (Schenk & Gärtner 2004, Schenk & Gärtner 2006).

7.1 Multiphase flows

To verify the implementation for multiphase flows we show results for the convected density jump in Section 7.1.1 and for the two-phase Poiseuille flow in Section 7.1.2. While for the first test case the densities in both phases are different, for the second test case the viscosity is varied. For the latter one we will show also results using a sharp interface, which is aligned with the grid edges. The equations for multiphase flows (2.8a)-(2.8c) have been discussed in Section 2.2.

7.1.1 Convected density jump

The first multiphase test case is the convection of a smooth density jump. It is an unsteady test case in one-dimensional space. Our simulations are extended to 2D by using periodic boundary conditions in the cross-stream direction. In Fig. 7.1 the computational grid and the initial condition for the level set function

$$\varphi(x_1, t=0) = \frac{1}{2} - \frac{1}{2}\tanh(20(x_1 - x_{\mathcal{I}}(t=0))) \tag{7.1}$$

are shown. In (7.1) the initial interface position is defined by $x_{\mathcal{I}}(t=0) = 0.2$. The computational domain is $\Omega = [0,1] \times [-0.1, 0.1]$. For velocity and pressure the solution at $t = 0$ is given by

$$u_1(x_1, t=0) = 1, \tag{7.2a}$$
$$u_2(x_1, t=0) = 0, \tag{7.2b}$$
$$p(x_1, t=0) = 0. \tag{7.2c}$$

At the left boundary of the domain Dirichlet values for both velocity components and the level set are prescribed, cf. (2.11a). A pressure outlet condition (2.11d) is used at

the right boundary, where the pressure is set to $p = 0$. The analytical solution for this test case is simply the pure convection of the initial density jump imposed by the level set function

$$\varphi(x_1, t) = \frac{1}{2} - \frac{1}{2}\tanh(20(x_1 - x_{\mathcal{I}}(t = 0) - t)). \tag{7.3}$$

For velocity and pressure there is no change over time. This test case is inspired by (Hahn 2009), (Olbricht 2009) and (Rauwoens et al. 2009), where similar examples were simulated in the context of the FVM testing various pressure-correction schemes designed for the application to combustion problems. For the density ratio between the two phases we chose $\rho_1/\rho_2 = 1000$, which is approximately the ratio of water to air. Such a high density ratio is quite a challenging test for variable density solvers. Even though this test case is purely determined by convection, we solve the complete system of equations (2.8a)-(2.8c) including diffusion in the momentum equation [1]. The viscosity in both faces is constant $\mu_1 = \mu_2 = 1.0$ and the Reynolds number is set to $Re = 10$.

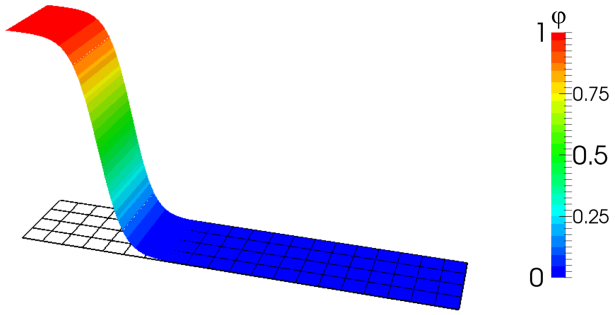


Figure 7.1: Computational grid with 4×20 cells and initial condition of level set function for convected density jump. The computational domain is $\Omega = [0, 1] \times [-0.1, 0.1]$.

For the spatial discretization we use a uniform Cartesian grid with 4×20 cells and polynomial orders of 3 for velocity and level set and 2 for pressure, which results in approximately 4-5 cells over the density jump in the streamwise direction, cf. Fig. 7.1. The BDF-2 scheme is applied for the temporal discretization with a time step size of $\Delta t = 5 \times 10^{-4}$. We run the simulation until $t_E = 0.6$, i.e. the final interface position is at $x_{\mathcal{I}}(t_E) = 0.8$. For the SIMPLE algorithm we apply the diagonal option, cf. Eq. (6.47), and the relaxation factors for velocity and pressure are set to $\alpha_u = 0.7$ and $\alpha_p = 0.3$. For the level set no under-relaxation is used. The convergence criteria for the pressure correction and the changes in the velocity and the level set are set to 10^{-6}.

The results for the level set function, the error of the streamwise velocity u_1 with respect to the exact solution $u_1 = 1$ and the pressure versus the streamwise direction

[1] We also run simulations for the convected density jump, where the viscous terms in the momentum equations were deactivated. These simulations were also stable and there are no significant differences compared to the presented results including diffusion in the momentum equations.

for different points in time are plotted in Fig. 7.2. The advection of the level set is exactly captured. There is no recognizable numerical diffusion. The error in the velocity is localized at the region of the density jump and returns to zero after the density jump has moved further downstream. The same applies for the pressure, which has got a nonzero gradient only close to the density jump. To the left and the right of the density jump the pressure is constant again. The errors in the velocity and the pressure gradient at the density jump result from temporal and spatial discretization errors, which will be confirmed in the next Section 7.1.1.1 by a convergence study for different grids and polynomial orders. At this point, we can already state that the implemented algorithm is stable for this challenging test case of a convected density jump with a ratio of $\rho_1/\rho_2 = 1000$.

7.1.1.1 Spatial convergence study

Now, we examine the convergence rates under h-refinement for various polynomial orders. For velocity and level-set the DG order is varied from 1 to 3 and accordingly for pressure from 0 to 2. Three different Cartesian grids with 8×40, 10×50 and 12×60 cells are used. To assure that the spatial error is dominating, a time steps size of $\Delta t = 10^{-4}$ is taken and the BDF-4 scheme is applied. The initial position of the interface is set to $x_{\mathcal{I}}(t_E) = 0.6$. For the convergence study a time period of 0.1 is simulated, i.e. the final position of the interface is $x_{\mathcal{I}}(t_E) = 0.7$. The SIMPLE iterations in each time step are terminated when the specified convergence criterion of 10^{-7} is reached. In Fig. 7.3 the results of the h-convergence study for the streamwise velocity u_1, the pressure and the level set are displayed. The observed convergence rates of the velocity and the pressure are slightly higher than expected, i.e. the rate for the velocity is at least $k+1$ and for the pressure it is also higher than $k'+1$. To determine the convergence rates more precisely further tests with different grids and polynomial orders would be necessary. For the level set we find an experimental order of convergence, which agrees almost exactly with the expected value of $k+1$. In summary, the qualitative results for the convected density jump of the last section and this convergence study shows the correct implementation of the unsteady multiphase solver with variable density. In Section 7.1.2, we will present results for a test case with different viscosities in both phases.

7.1.1.2 Performance

The performance of the SIMPLE algorithm for this test case is demonstrated by the convergence history shown in Fig. 7.4, where the pressure correction versus the number of SIMPLE iterations is plotted for the first time steps. The performance is compared for the calculations of the last section on the mesh with 12×60 cells for varying polynomial order. First of all, we observe an almost constant convergence rate for the pressure correction down to the specified convergence criterion (the convergence rates for the velocity and the level set, which are not shown, are nearly identical). The number of SIMPLE iterations in each time step is hardly influenced by the polynomial

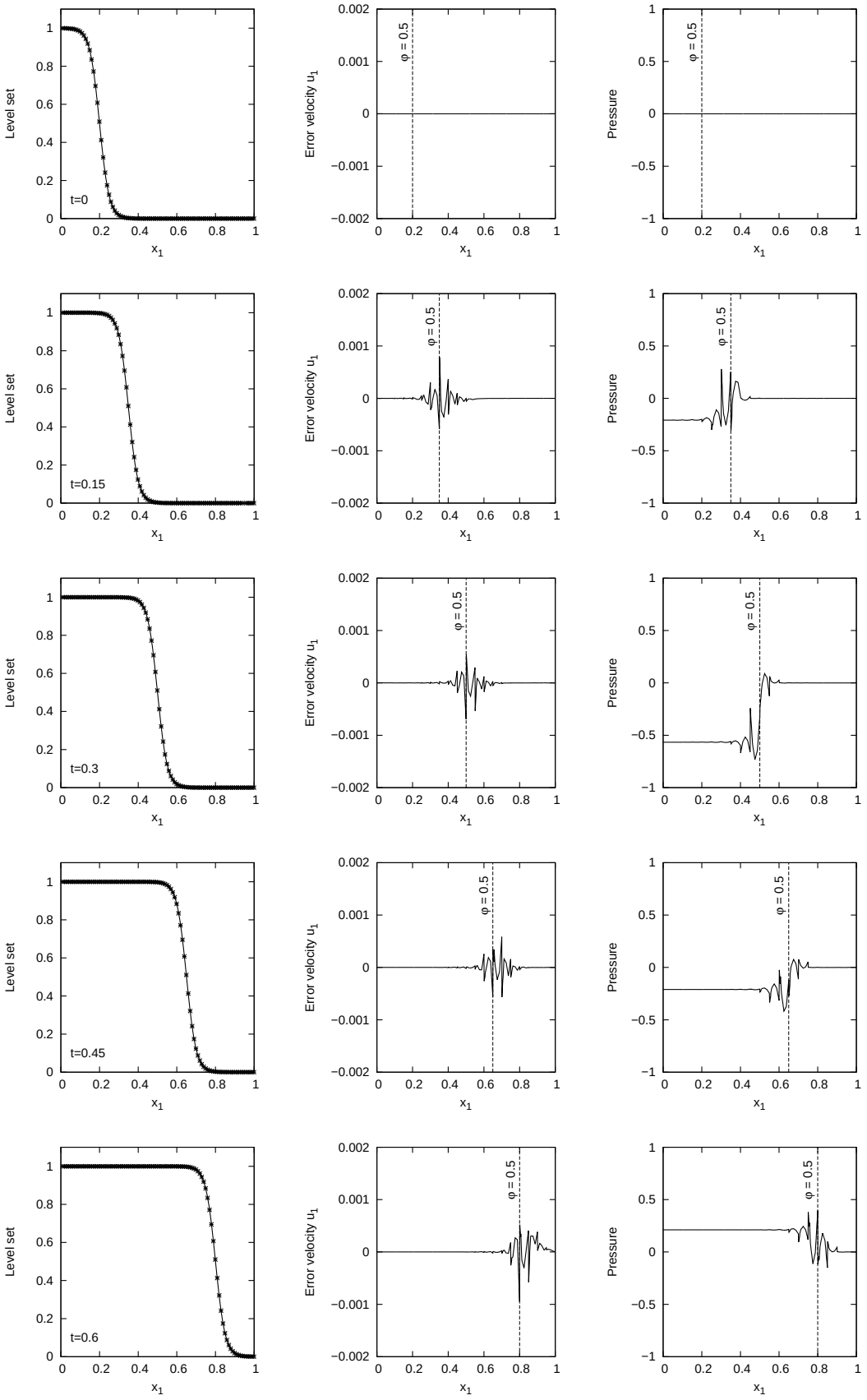


Figure 7.2: Level set, error of velocity u_1 and pressure versus streamwise coordinate at different points in time for the convected density jump. The calculated results for the level set (points) are plotted together with the analytical solution (lines). For the plots of the velocity error and the pressure the current interface position is indicated by the dashed line marked with $\varphi = 0.5$.

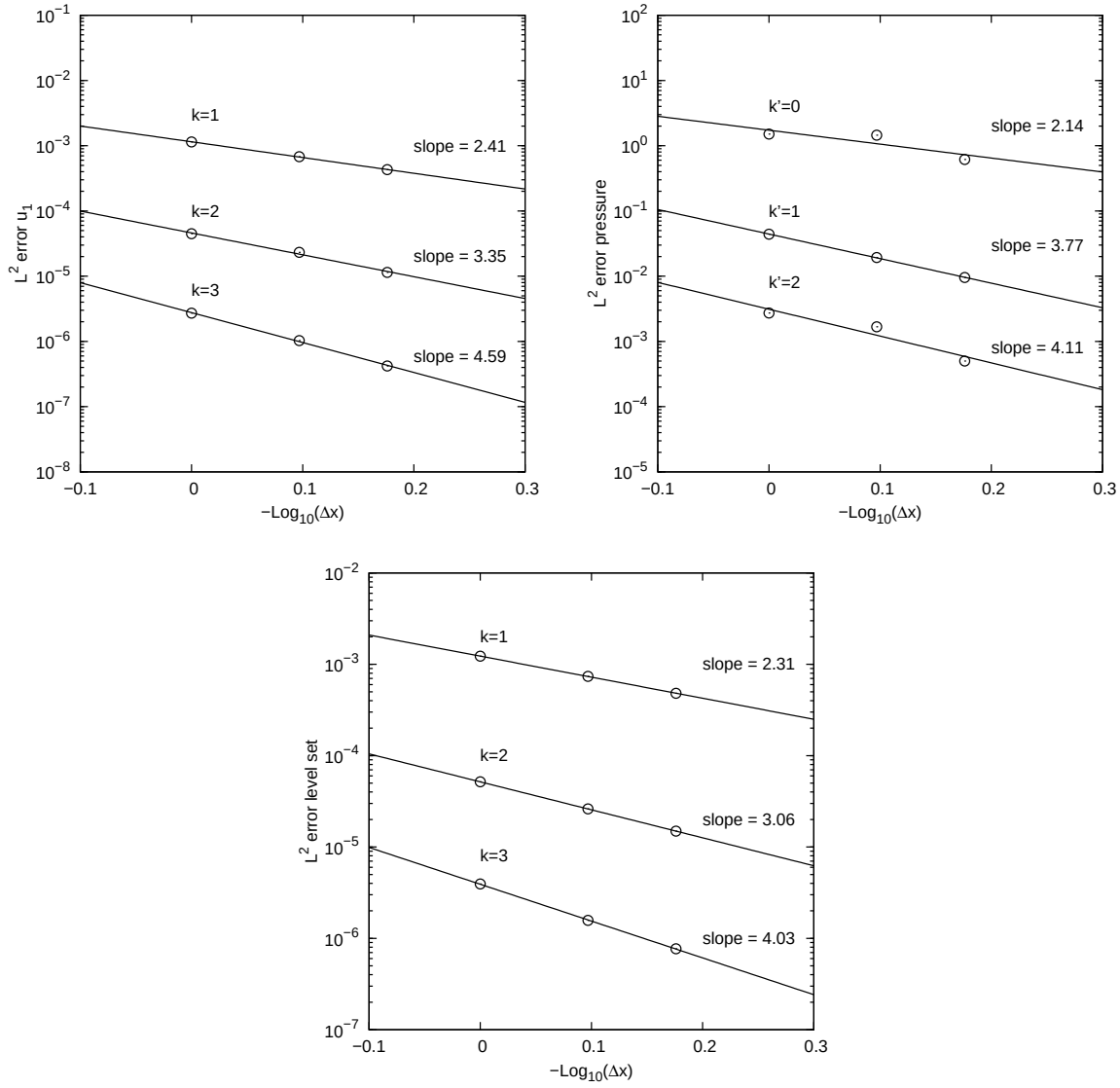


Figure 7.3: h-convergence study for convected density jump. L^2 errors of streamwise velocity (top left), pressure (top right) and level set (bottom) versus normalized grid size and linear fits with corresponding slopes.

order. On average, about 90 SIMPLE iterations per time step are needed to reach the specified convergence criterion.

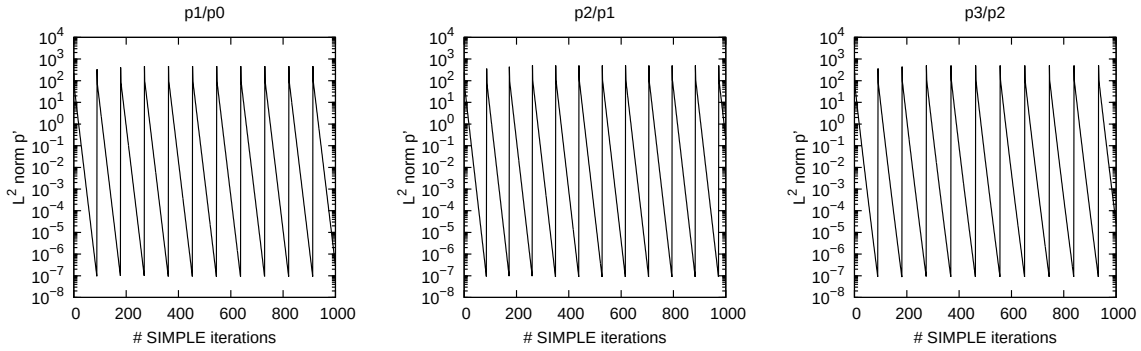


Figure 7.4: Performance of SIMPLE algorithm for convected density jump. The pressure correction is plotted versus the number of SIMPLE iterations for the first time steps using the diagonal option. The grid consists of 12×60 cells. The performance is compared for different polynomial orders p1/p0, p2/p1 and p3/p2, where the first number indicates the DG order for velocity and level set and the second number for pressure.

7.1.2 Two-phase Poiseuille flow

To test the multiphase solver with variable viscosity, we simulate the two-phase Poiseuille flow, which is a two-dimensional channel flow with a horizontal interface at the center line of the channel. The test case is completely defined by the following parameters:

- The Reynolds number, which we set to $Re = 10$.
- The viscosity ratio of the two phases is defined as $\mu_2/\mu_1 = 10$.
- The pressure gradient in the streamwise direction is assumed to be $\frac{\partial p}{\partial x_1} = -1$.
- The density is set to a constant value, i.e. $\rho_1 = \rho_2 = 1$.

Then, the analytical solution for velocity and pressure will take the form

$$u_1 = u_1(x_2), \tag{7.4}$$
$$u_2 = 0, \tag{7.5}$$
$$p = -x_1 + C, \tag{7.6}$$

where $C \in \mathbb{R}$ is an arbitrary constant. Using (7.4)-(7.6) the continuity equation and the momentum equation in the cross-stream direction are identically fulfilled. The momentum equation in the streamwise direction reduces to

$$\frac{1}{Re}\frac{\partial}{\partial x_2}\left(\mu\frac{\partial u_1}{\partial x_2}\right) = \frac{\partial p}{\partial x_1}. \tag{7.7}$$

7.1.2.1 Smooth interface

Now, we first assume a smooth interface represented by the following level set function

$$\varphi = \frac{1}{2} - \frac{1}{2}\tanh(10x_2). \tag{7.8}$$

Inserting the equation for the viscosity (2.6b) in (7.7) and integrating yields the analytical solution for the streamwise velocity

$$\begin{aligned} u_1 =& C_1 x_2 - 5x_2^2 - \frac{9}{200} C_1 \ln(1 + 10\exp(20x_2)) + \frac{9}{20} x_2 \ln(1 + 10\exp(20x_2)) \\ &+ \frac{9}{400}\,\mathrm{dilog}(-10\exp(20x_2)) + C_2, \end{aligned} \tag{7.9}$$

where dilog() is the dilogarithm function. The constants C_1 and C_2 in (7.9) can be determined by the no-slip boundary condition at the lower and upper wall.

To simulate this test case we solve the steady form of the equations (2.8a)-(2.8c) in the domain $\Omega = [0,5] \times [-1,1]$. At the left side of the domain boundary, Dirichlet boundary conditions are prescribed for the velocity and the level set by the analytical solution. An outflow boundary is used at the opposite domain boundary, cf. (2.11c), and the no-slip condition holds for the lower and upper wall, cf. (2.11b). Spatial discretization is done by a Cartesian grid with 10×25 cells and polynomials of order 2 are taken for velocity and level set and of order 1 for pressure. The initial values for the SIMPLE iterations for both velocity components and for the pressure are set to zero. The level set is initialized with the analytical solution (7.8). The block diagonal option for the approximation matrices in the SIMPLE algorithm is taken and the underrelaxation factors are set to $\alpha_u = 0.8$, $\alpha_p = 0.2$ and $\alpha_\varphi = 0.3$, where the relaxation for the level set is done in an explicit manner. The simulation is terminated when the specified convergence criterion of 10^{-6} is reached for all variables, i.e. velocity, pressure and level set.

In Fig. 7.5 the results of the simulation for the streamwise velocity and the level set are plotted along the vertical axis. The analytical solution is also shown. The calculated results for the velocity perfectly match the analytical solution. For the level set slight deviations between our result and the exact solution can be recognized close to the transition between the two phases, which can be explained by the rather coarse grids resulting in only two cells across the interface.

7.1.2.2 Sharp interface

Within the last years there is a very recent development extending the DGM to singular problems, where the solution might have kinks and jumps. One example of such problems is multiphase flows, where no regularization is used at the interface. Instead, the kinks and jumps at the interface are resolved with subcell accuracy, by either cutting

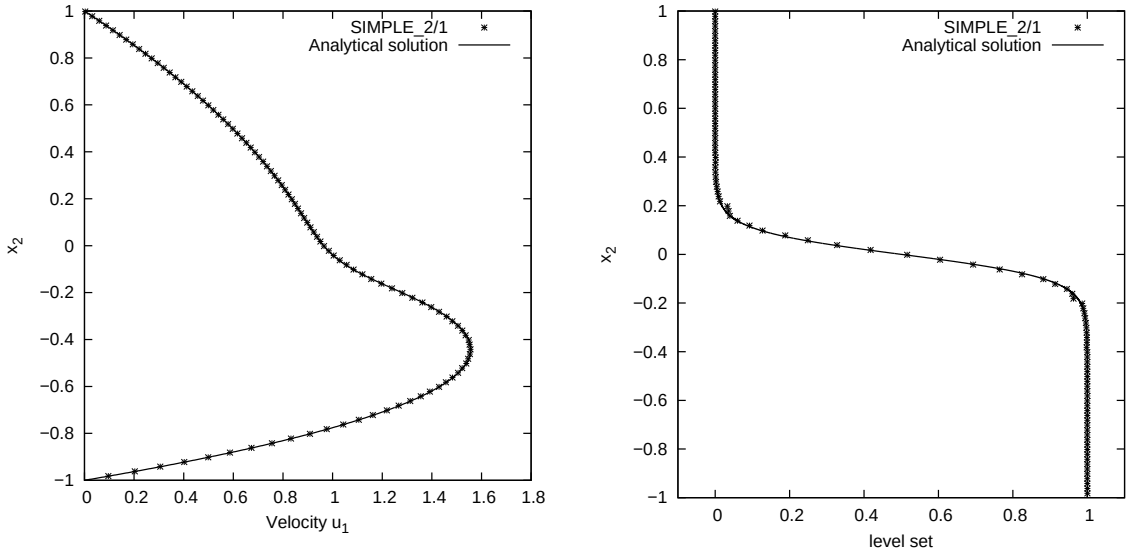


Figure 7.5: Velocity profile (left) and level set (right) for two-phase Poiseuille flow with smooth interface along the center line of the channel. The viscosity ratio is $\mu_2/\mu_1 = 10$.

the computational mesh at the interface or introducing special ansatz functions, cf. (Heimann et al. 2013), (Kummer & Oberlack 2013) and (Kummer 2013). To demonstrate the applicability of the SIMPLE algorithm in such a framework, we simulate the two-phase Poiseuille flow with a sharp interface, which is aligned with the grid edges. By doing so, kinks and jumps are naturally incorporated in the DGM. In that case, the analytical solution can be expressed using the Heaviside step function

$$H(-x_2) = \begin{cases} 1, & x_2 \leq 0, \\ 0, & x_2 > 0, \end{cases} \tag{7.10}$$

Taking into account the jump conditions for the streamwise velocity at the interface

$$[\![u_2]\!] = 0, \tag{7.11}$$

$$\left[\!\!\left[\mu \frac{\partial u_1}{\partial x_2}\right]\!\!\right] = 0, \tag{7.12}$$

the analytical solution with respect to the sharp interface is

$$u_2 = H(-x_2)\left(-5x_2^2 - \frac{45}{11}x_2 + \frac{10}{11}\right) + H(x_2)\left(-\frac{1}{2}x_2^2 - \frac{9}{22}x_2 + \frac{10}{11}\right). \tag{7.13}$$

Since we are not using any kind of extended DG method, we need to assure that the interface is exactly aligned with the grid edges. This is accomplished by prescribing the level set function (7.10) rather than calculating the solution for the level set, i.e. solving of the level set equation within the SIMPLE iterations is deactivated for this test.

The result for the streamwise velocity profile and the analytical solution are shown in Fig. 7.6. The kink at the interface is exactly reproduced by the numerical solution. Moreover, since the exact solution (7.13) for the velocity is a polynomial function of second order in both phases and the exact pressure is a linear function, the numerical solution using the DG order 2 for velocity and 1 for pressure is exact up to machine accuracy.

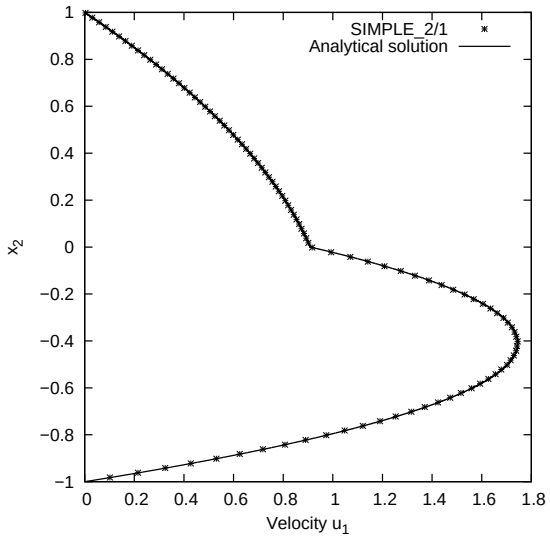


Figure 7.6: Velocity profile for two-phase Poiseuille flow with sharp interface along the center line of the channel. The viscosity ratio is $\mu_2/\mu_1 = 10$.

In Fig. 7.7 the convergence history of the SIMPLE algorithm is shown using relaxation factors of $\alpha_u = 0.9$ and $\alpha_p = 0.2$. The convergence rate for the pressure correction is almost constant down to machine accuracy. It can also be seen that the errors of both velocity components and the pressure with respect to the exact solution are reduced to machine accuracy, which confirms the correct representation of the kink at the interface.

At the end of this section, we conclude that the implementation of the multiphase solver is also correct for different viscosities in both phases. Additionally, we could show that the SIMPLE algorithm also performs well for settings with a sharp interface like encountered in extended DG methods.

7.2 Low-Mach number flows

The implementation in BoSSS of the solver for low-Mach number flows is verified by simulating three different test cases. The first one is the Couette flow with a vertical temperature gradient in Section 7.2.1. This test case is simulated with constant dynamic viscosity and thermal conductivity as well as variable properties using a power law. In both cases analytical solutions are available, which are used to perform h-convergence

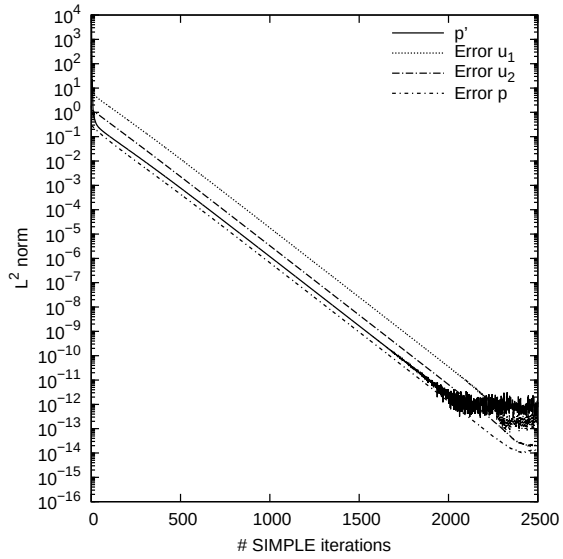


Figure 7.7: Performance of SIMPLE algorithm for two-phase Poiseuille flow with sharp interface. Plotted are the pressure correction and the errors of both velocity components and pressure with respect to the analytical solution versus the number of SIMPLE iterations.

studies. The second test case in Section 7.2.2 is the natural convection in a square cavity, which is a common benchmark problem for low-Mach number solvers. The Rayleigh number is varied between $Ra = 10^2$ and $Ra = 10^6$. All these Rayleigh numbers yield steady solutions. The last test case in Section 7.2.3 is the natural convection in a tall cavity with a height to width aspect ratio of 8:1. This test case is simulated for a Rayleigh number above a critical value, where an unsteady time-periodic solution is obtained, thus testing the solver for unsteady low-Mach number flows.

7.2.1 Couette flow with temperature gradient

The first test case to verify the implementation of the low-Mach number solver is the Couette flow with a vertical temperature gradient. The geometry and boundary conditions are shown in Fig. 7.8. The upper wall is moving with prescribed velocity $u_1 = 1$ and temperature $T = T_h$ and at the lower wall $u_1 = 0$ and $T = T_c$ with $T_c < T_h$. At the left and right boundaries of the domain $\Omega = [0,1] \times [0,1]$ Dirichlet boundary conditions for velocity and temperature are used. The direction of gravity is aligned with the vertical axis.

The solution for this setting will be of the following form

$$u_1 = u_1(x_2), \tag{7.14a}$$
$$u_2 = 0, \tag{7.14b}$$
$$p = p(x_2), \tag{7.14c}$$
$$T = T(x_2). \tag{7.14d}$$

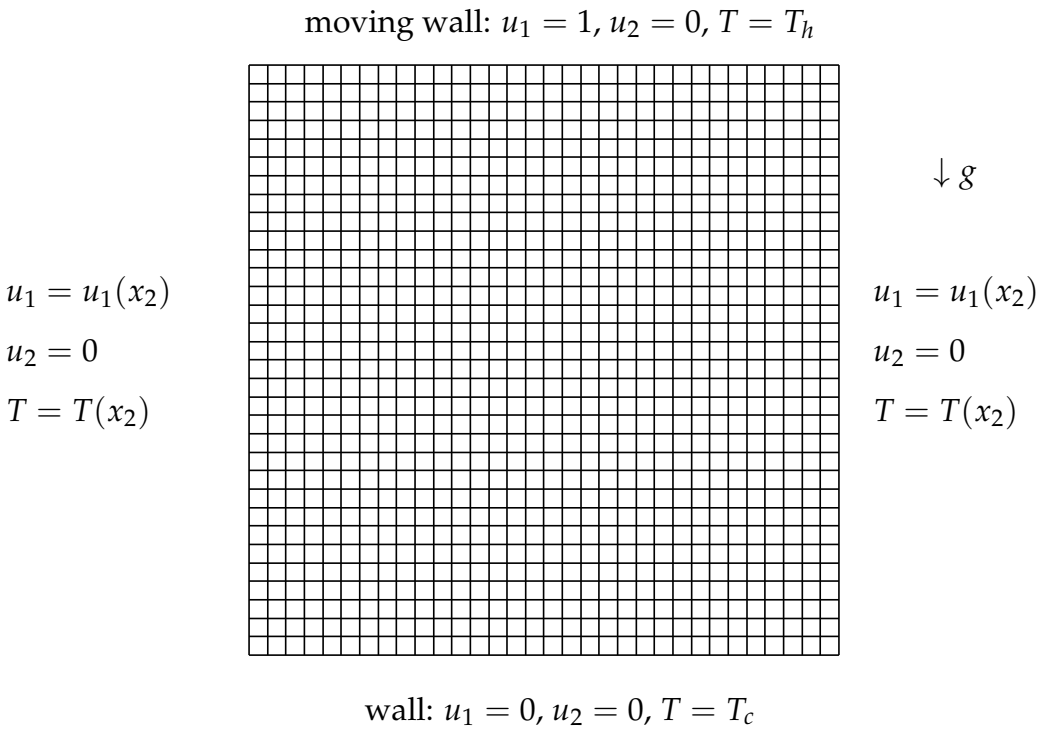


Figure 7.8: Geometry and computational grid with 32×32 cells for Couette flow with temperature gradient. The domain is $\Omega = [0,1] \times [0,1]$ and Dirichlet boundary conditions are given on all domain boundaries.

Using (7.14a)-(7.14d) the continuity equation (2.14b) is identically fulfilled and the momentum (2.14c) and temperature (2.14d) equations reduces to

$$\frac{1}{Re}\frac{\partial}{\partial x_2}\left(\mu\frac{\partial u_1}{\partial x_2}\right) = 0, \tag{7.15a}$$

$$\frac{\partial p}{\partial x_2} = -\frac{1}{Fr^2}\frac{p_0}{T}, \tag{7.15b}$$

$$\frac{1}{RePr}\frac{\partial}{\partial x_2}\left(\lambda\frac{\partial T}{\partial x_2}\right) = 0, \tag{7.15c}$$

where $\rho = p_0/T$ is used in (7.15b). Assuming constant viscosity and thermal conductivity, we obtain the analytical solution

$$u = x_2, \tag{7.16a}$$

$$p = -\frac{p_0}{Fr^2(T_h - T_c)}\ln\left((T_h - T_c)x_2 + T_c\right) + C, \tag{7.16b}$$

$$T = (T_h - T_c)x_2 + T_c, \tag{7.16c}$$

where $C \in \mathbb{R}$ is an arbitrary constant. By taking the power law to determine the viscosity

$$\mu = T^{\frac{2}{3}}, \tag{7.17}$$

and for constant Pr number the analytical solution changes to

$$u = C_1 + C_2 \left(x_2 + \frac{T_c^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5}, \tag{7.18a}$$

$$p = -\frac{5}{2} \frac{p_0 \left(x_2 \left(T_h^{5/3} - T_c^{5/3} \right) + T_c^{5/3} \right)^{2/5}}{Fr^2 \left(T_h^{5/3} - T_c^{5/3} \right)} + C_3, \tag{7.18b}$$

$$T = \left(C_4 - \frac{5}{3} C_5 x_2 \right)^{3/5}. \tag{7.18c}$$

In (7.18b) C_3 is the arbitrary constant for the pressure and the constants C_1 and C_2 in (7.18a) for the velocity and C_4 and C_5 for the temperature in (7.18c) are determined by the boundary conditions at the lower and upper wall

$$C_1 = \frac{\left(\frac{T_c^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5}}{\left(\frac{T_c^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5} - \left(\frac{T_h^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5}}, \tag{7.19a}$$

$$C_2 = \frac{1}{\left(\frac{T_h^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5} - \left(\frac{T_c^{5/3}}{T_h^{5/3} - T_c^{5/3}} \right)^{3/5}}, \tag{7.19b}$$

$$C_4 = T_c^{5/3}, \tag{7.19c}$$

$$C_5 = \frac{3}{5} \left(T_c^{5/3} - T_h^{5/3} \right). \tag{7.19d}$$

In the following two sections the derived analytical solutions (7.16a)-(7.16c) and (7.18a)-(7.18c) will be used to test the h-convergence of the low-Mach number solver. The dimensionless parameters are set to $Re = 10$, $Pr = 0.71$, $p_0 = 1.0$, $T_h = 1.6$, $T_c = 0.4$. The Froude number is calculated in the same manner like in Section 7.2.2

$$Fr = \left(\frac{2Pr(T_h - T_c)}{(T_h + T_c)} \right)^{\frac{1}{2}}, \tag{7.20}$$

which yields $Fr = 0.92303846$. In Fig. 7.9 the analytical solution for velocity, pressure and temperature taking the power law is plotted.

7.2.1.1 Constant viscosity

First, we study the h-convergence for constant viscosity and thermal conductivity. The domain is discretized by uniform Cartesian grids with 4×4, 8×8, 16×6 and 32×32 cells. Polynomial orders are varied from 1 to 5 for velocity and temperature and 0 to 4 for pressure. The analytical solution (7.16a)-(7.16c) is used to define Dirichlet boundary conditions for velocity and temperature on all domain boundaries. For the pressure a

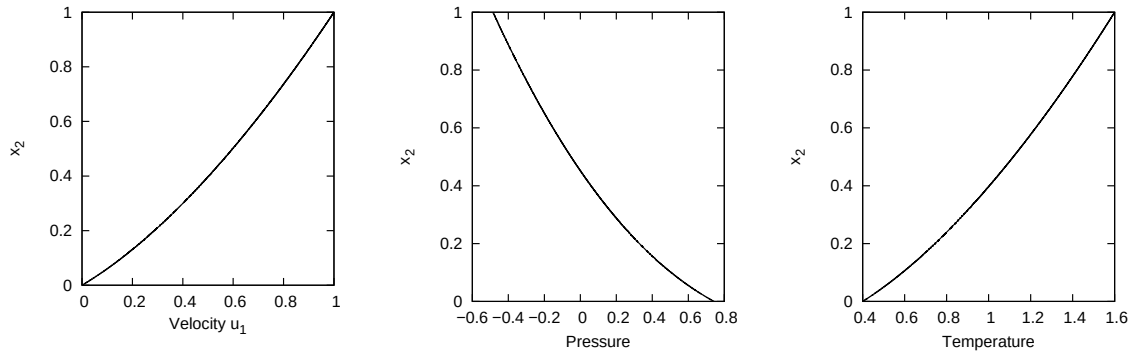


Figure 7.9: Analytical solution for Couette flow with temperature gradient using the power law for viscosity.

reference point has to be set. The analytical solution is also used as initial guess for the SIMPLE iterations, which are terminated when the specified convergence criteria of $\|p'_h\|_{L^2(\Omega)} < 10^{-10}$, $\|u_{hi}^{\vartheta+1} - u_{hi}^{\vartheta}\|_{L^2(\Omega)} < 10^{-12}$ and $\|T_h^{\vartheta+1} - T_h^{\vartheta}\|_{L^2(\Omega)} < 10^{-12}$ are reached. Relaxation factors are set to $\alpha_p = 0.5$ and $\alpha_u = 0.8$. For the temperature no under-relaxation is used. The block diagonal option is used for the approximation in the correction step of the SIMPLE algorithm. In Fig. 7.10 the results of the h-convergence study are shown. For both velocity components and the temperature we observe convergence rates in the range of $k+1$ and for the pressure of approximately $k'+1$. We note that the errors on the finest grid for the polynomial orders $k = 5$ and $k' = 4$ in Fig. 7.10 have been omitted, since the errors for velocity and temperature are close to machine accuracy.

7.2.1.2 Power-law viscosity

Next, we investigate the h-convergence for non-constant viscosity using the power law (7.17). The thermal conductivity is given by assuming constant Prandtl number. The settings for the SIMPLE algorithm are the same like in the previous section for constant viscosity. To observe the expected rates of convergence finer grids have to be used for this more complicated test case. Again, uniform Cartesian grids with 16×16, 24×24, 32×32, 40×40, 48×48, 56×56 and 64×64 cells are used. The polynomial orders for velocity and temperature are varied from 1 to 3 and for pressure from 0 to 2. In Fig. 7.11 the h-convergence study is depicted. We observe similar convergence rates like for the constant viscosity test case, i.e. approximately $k+1$ for velocity and temperature and $k'+1$ for pressure.

7.2.2 Natural convection in heated cavity

A common benchmark problem for low-Mach number solvers is the natural convection in an enclosed cavity. For the definition of the problem see Fig. 7.12. The fluid in the square cavity is subjected to a hot wall with temperature T_h at the left side of the

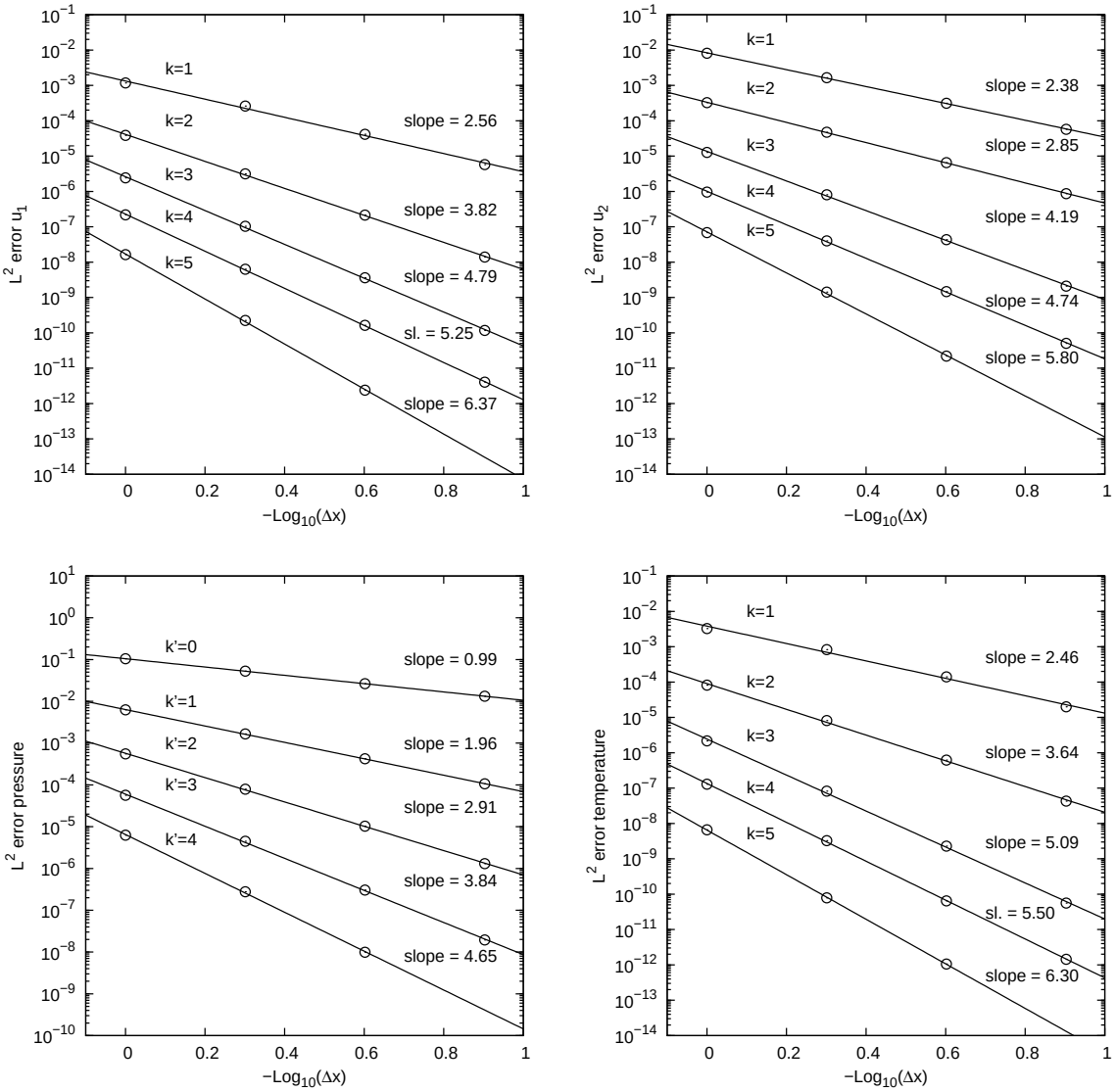


Figure 7.10: *h*-convergence study for Couette flow with temperature gradient assuming constant viscosity. L^2 errors versus normalized grid size and linear fits with corresponding slopes.

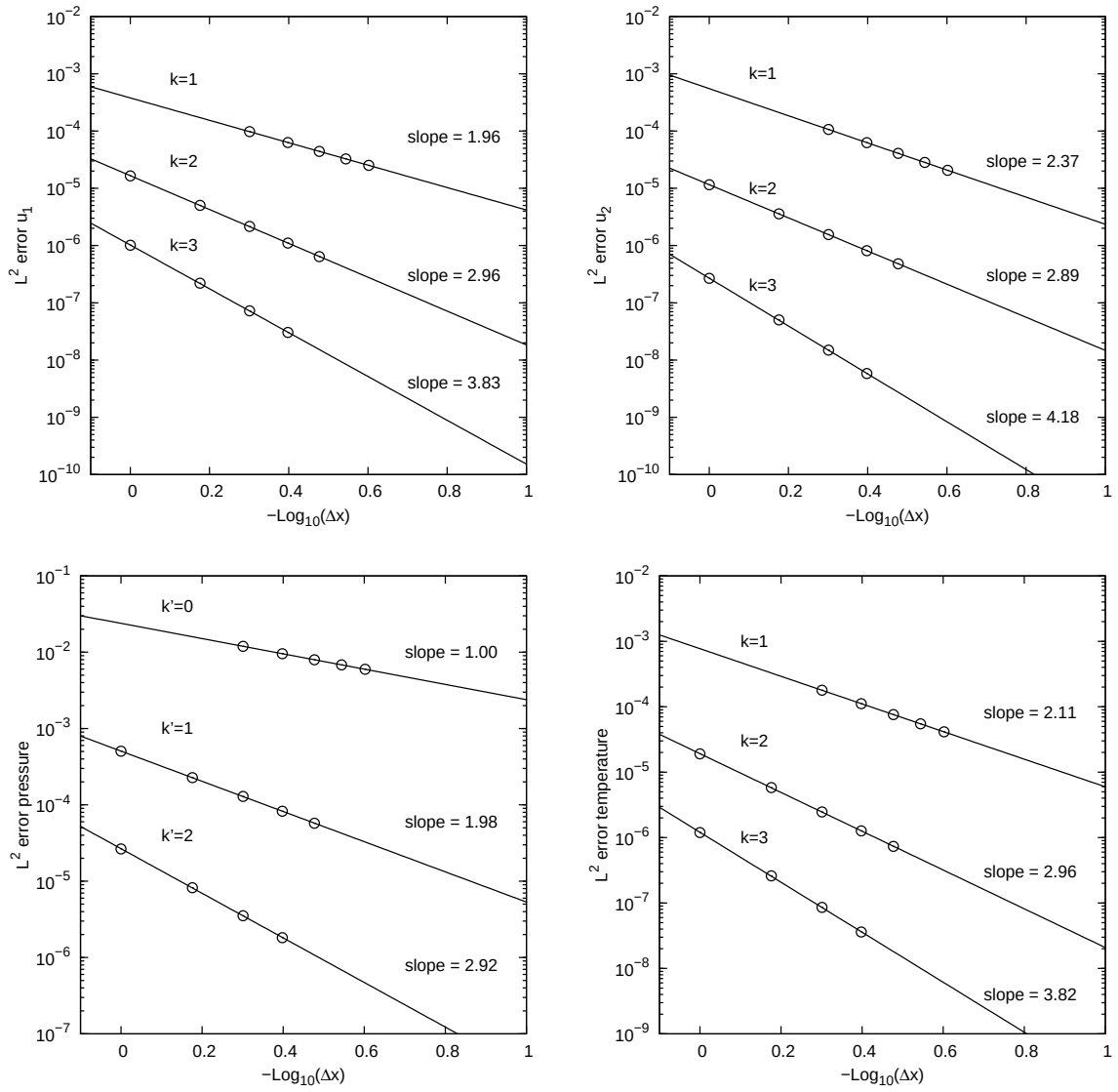


Figure 7.11: *h*-convergence study for Couette flow with temperature gradient using power law for viscosity. L^2 errors versus normalized grid size and linear fits with corresponding slopes.

domain boundary and a cold wall with temperature T_c on the right side. The top and bottom boundaries are adiabatic walls. To all domain boundaries the no-slip condition is imposed. The gravity, which is directed downwards, will induce a pressure field, which in turn will initiate a circulating flow in clockwise direction.

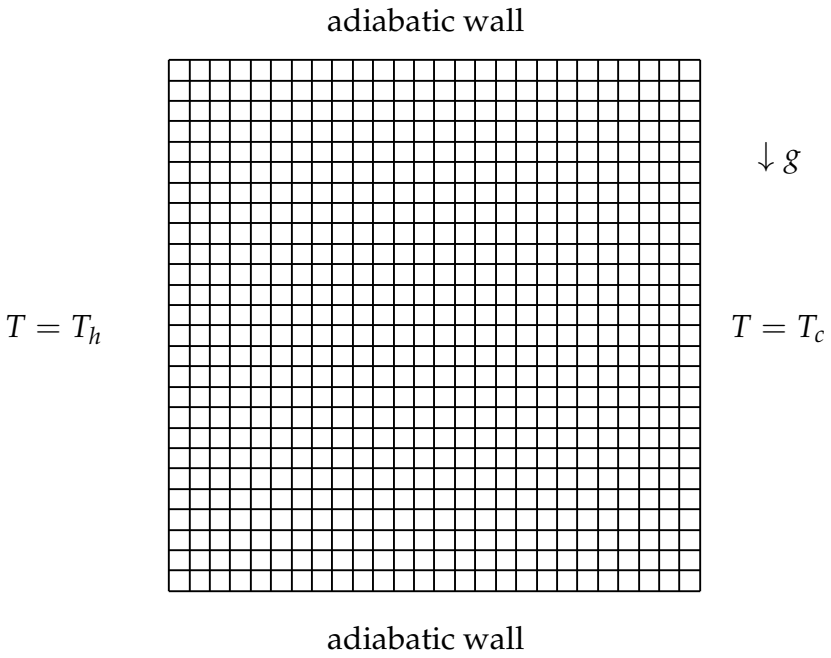


Figure 7.12: Geometry and computational grid with 26×26 cells for natural convection in heated cavity. The domain is $\Omega = [0,1] \times [0,1]$. The top and bottom boundaries are adiabatic walls and at the left and right boundaries the temperature is prescribed. The no-slip condition applies to all domain boundaries.

In (De Vahl Davis & Jones 1983) and (De Vahl Davis 1983) this test case was originally defined for small temperature differences $T_h - T_c$, where the Boussinesq approximation is valid. Later, due to a lack of reference solutions for non-Boussinesq natural convection flows this test case was extended to larger temperature differences, see (Le Quéré, Weisman, Paillère, Vierendeels, Dick, Becker, Braack & Locke 2005) and (Paillère, Le Quéré, Weisman, Vierendeels, Dick, Braack, Dabbene, Beccantini, Studer, Kloczko, Corre, Heuveline, Darbandi & Hosseinizadeh 2005). Extensive benchmark data from various groups using different codes and modeling approaches, e.g. low-Mach number codes (Becker & Braack 2002) as well as fully compressible models (Vierendeels, Merci & Dick 2003), are provided. For our studies we take the results from (Vierendeels et al. 2003) as reference, as it contains the most comprehensive data and was also found to be one of the most accurate solutions, cf. (Le Quéré et al. 2005, Paillère et al. 2005). The benchmark solutions in (Vierendeels et al. 2003) are calculated based on the fully compressible Navier-Stokes equations, i.e. no low-Mach number approximation is used. Spatial discretization is done by a FVM of second order accuracy on a stretched grid with 1024×1024 cells.

Using the low-Mach number approximation, the setup for this benchmark is completely defined by the following parameters, cf. (Vierendeels et al. 2003):

- The Prandtl number is set to $Pr = 0.71$ and the heat capacity ratio to $\kappa = 1.4$.
- The Rayleigh number

$$Ra = Pr\frac{\tilde{g}\tilde{\rho}_\infty^2(\tilde{T}_h - \tilde{T}_c)\tilde{L}^3}{\tilde{T}_\infty\tilde{\mu}_\infty^2}, \tag{7.21}$$

  where $\tilde{T}_\infty = (\tilde{T}_h + \tilde{T}_c)/2$. For small values of Ra heat transfer is dominated by conduction, whereas for larger values of Ra convection is dominating, cf. Fig. 7.13. Using the definition for the reference velocity from (Vierendeels et al. 2003)

$$\tilde{u}_\infty = \frac{Ra^{1/2}\tilde{\mu}_\infty}{\tilde{\rho}_\infty\tilde{L}}, \tag{7.22}$$

  yields the following relation between Reynolds and Rayleigh number

$$Re = Ra^{1/2}, \tag{7.23}$$

  which is used to adjust the setup of our simulations for the different Rayleigh numbers, where the Reynolds number has to be prescribed rather than the Rayleigh number.
- The temperature difference expressed in terms of

$$\epsilon = \frac{\tilde{T}_h - \tilde{T}_c}{2\tilde{T}_\infty}. \tag{7.24}$$

  We will show results and compare to the reference solution for $\epsilon = 0.6$ and $\tilde{T}_\infty = 600K$. Using the definitions above, the Froude number can be evaluated from

$$Fr = (Pr2\epsilon)^{1/2}, \tag{7.25}$$

  and the dimensionless values for the temperature at the hot and cold wall are given by

$$T_h = 1.6, \tag{7.26a}$$
$$T_c = 0.4. \tag{7.26b}$$

- The parameter S to define the Sutherland's law is set to $S = 110.5K$.
- The initial mass needs to be specified to determine the thermodynamic pressure

$$p_0 = \frac{m(t=0)}{\int \frac{1}{T} dV}. \tag{2.21, repeated}$$

  The reference values for nondimensionalization are chosen such that the dimensionless initial mass is $m(t = 0) = 1.0$.

For $\epsilon = 0.6$ and $\tilde{T}_\infty = 600K$ benchmark data are available for six different Rayleigh numbers $Ra = \{10^2, 10^3, 10^4, 10^5, 10^6, 10^7\}$. Since all Rayleigh numbers are in the steady regime, we use the steady solver. The initial guess for the SIMPLE iterations is given by

$$u_1^0 = 0, \tag{7.27a}$$
$$u_2^0 = 0, \tag{7.27b}$$
$$p^0 = -\frac{1}{Fr^2}\rho^0 x_2, \tag{7.27c}$$
$$T^0 = 1, \tag{7.27d}$$

where the pressure is initialized according to the hydrostatic pressure field for the given initial density distribution, which is $\rho^0 = 1$. A reference point for the pressure has to be set, since there is no boundary condition, which determines the pressure level. In each step of the SIMPLE algorithm the thermodynamic pressure is rescaled according to (2.21).

In Section 7.2.2.1 we show contours and profiles for velocity and temperature and compare the results to the benchmark data. The results are obtained using a uniform Cartesian grid with 26×26 cells and polynomial orders of 5 for velocity and temperature and 4 for pressure. The convergence criteria for the pressure correction and the change in the velocity components and the temperature are set to 10^{-10}. In Table 7.1 the relaxation factors for velocity and pressure for the different Rayleigh numbers can be found, where we use the block diagonal option for the approximation in the corrector step, cf. Eq. (6.48). To get a convergent solution for $Ra = 10^7$ more under-relaxation for velocity is needed compared to the test cases with lower Rayleigh numbers. For the temperature no under-relaxation is used.

Table 7.1: Relaxation factors for velocity and pressure for natural convection in heated cavity using the block diagonal option. For temperature no under-relaxation is used.

Ra	10^2	10^3	10^4	10^5	10^6	10^7
α_u	0.9	0.9	0.9	0.9	0.9	0.6
α_p	0.2	0.2	0.2	0.2	0.1	0.4

In Section 7.2.2.2 we study the accuracy with respect to the Nusselt number and thermodynamic pressure for different grids and polynomial orders. Finally, in Section 7.2.2.3 we show some results for the performance of the SIMPLE algorithm for this test case.

7.2.2.1 Contours and profiles of temperature and velocity

We begin by discussing the temperature profiles for all six Rayleigh numbers shown in Fig. 7.13. It can be observed that for $Ra = 10^2$ heat transfer is dominated by

conduction, the temperature profile is almost linear. For $Ra = 10^3$ the starting influence of convection on the temperature profile can be seen clearly. When increasing the Rayleigh number further, the convection is getting more dominant until most of the temperature isolines are almost horizontal for $Ra = 10^7$. In Fig. 7.14 the corresponding streamlines are shown. At a Rayleigh number of $Ra = 10^2$ there is one vortex and the flow field is almost symmetrical. For $Ra = 10^3$ and $Ra = 10^4$ the vortex center is shifted down towards the right corner. A second vortex can be observed for $Ra = 10^5$. Increasing the Rayleigh number to $Ra = 10^6$ the flow pattern is getting more complex and for $Ra = 10^7$ a counter rotating vortex can be found close to the bottom right corner. Comparing visually the temperature profiles and streamlines to the benchmark solution no difference can be recognized.

A more detailed comparison can be done by plotting the temperature and velocity components along some predefined lines. The temperature is evaluated at the vertical line where $x_1 = 0.5$ and at three horizontal lines with $x_2 = 0$, $x_2 = 0.5$ and $x_2 = 1$. The velocity component u_1 is plotted along the vertical line at $x_1 = 0.5$ and u_2 at the horizontal line $x_2 = 0.5$. To compare our results with the benchmark solution, we extract data points from the figures in (Vierendeels et al. 2003) using the software g3data[2]. In Fig. 7.15 the temperature profiles along the vertical line and in Fig. 7.16 along the three horizontal lines for the different Rayleigh numbers are plotted. The agreement with the reference data is excellent. Just for $Ra = 10^7$ minor deviations are recognizable for the temperature profile along the horizontal line at $y = 0$ close to the hot and cold wall, cf. Fig. 7.16. Considering the coarse and uniform grid we use leading to a factor of about $1/70$ in the number of DOF compared to the reference solution[3] our results are quite accurate. Next, we look at the velocity profiles shown in Fig. 7.17 and 7.18. Again, there is no observable difference between our results and the benchmark solution. Even though our grid is rather coarse and uniform, the steep gradients in the profile of the velocity component u_2, cf. Fig. 7.18, at the hot and cold wall for the Rayleigh numbers $Ra = 10^5 - 10^7$ is perfectly captured.

7.2.2.2 Nusselt number and thermodynamic pressure

Next, we investigate the accuracy for different combinations of grids and polynomial orders with respect to the Nusselt number and the thermodynamic pressure. The Nusselt number characterizes the heat flux through the walls and is defined in the averaged form as

$$Nu = \frac{1}{T_h - T_c} \int_\Gamma \lambda \frac{\partial T}{\partial x_1} dx_2. \tag{7.28}$$

By integrating the temperature equation (2.14d) in its steady form over the computational domain Ω and using the no-slip condition for the velocity and the Neumann

[2] http://www.frantz.fi/software/g3data.php

[3] The number of DOF per variable are for the present results $\text{DOF}_{\text{present}} = 26^2 * 21 = 14,196$ and for the reference solution $\text{DOF}_{\text{ref}} = 1024^2 = 1,048,573$ leading to a factor $\text{DOF}_{\text{present}}/\text{DOF}_{\text{ref}} \approx 1/73.9$.

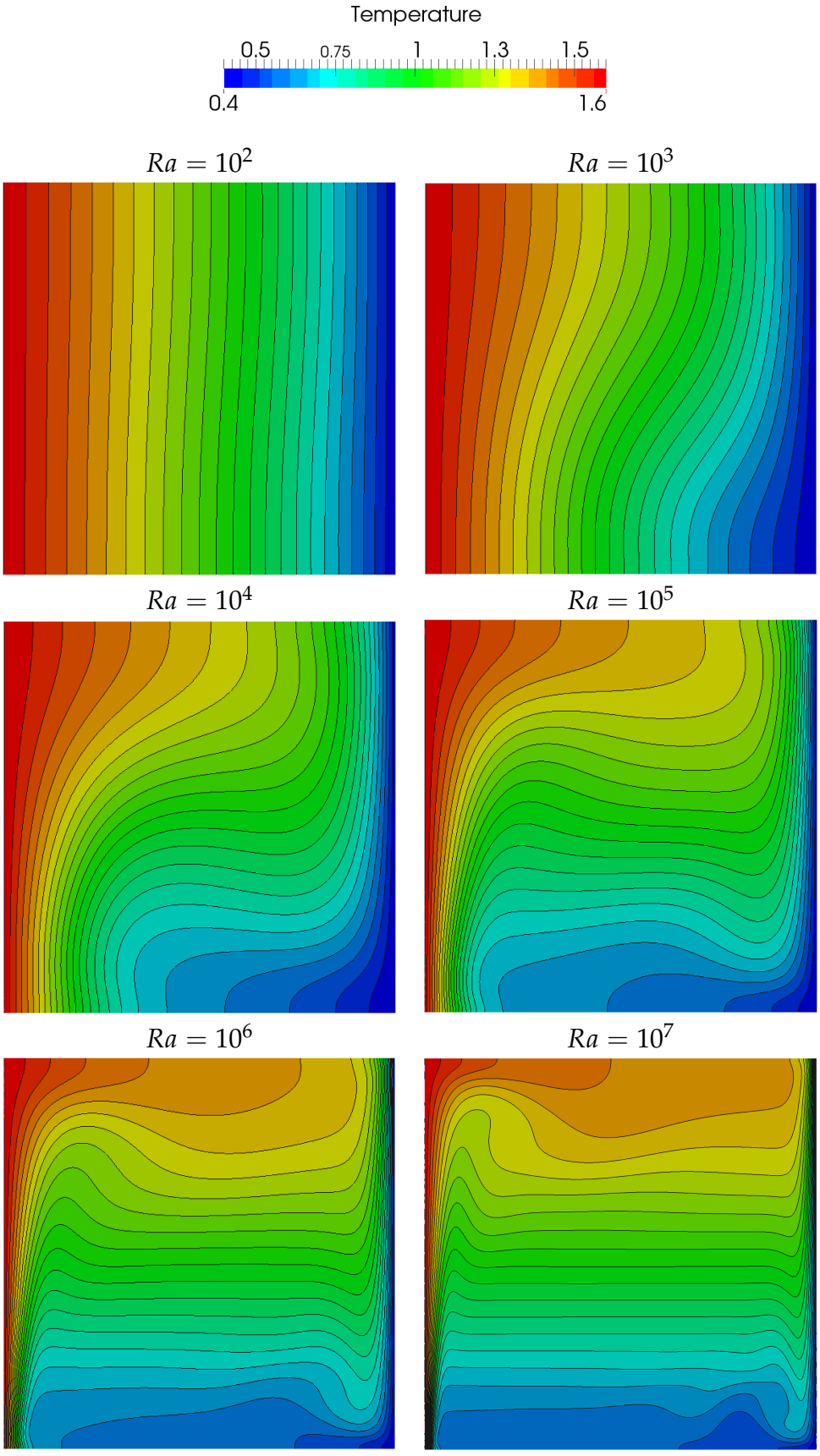


Figure 7.13: Temperature contours for $Ra = 10^2 - 10^7$ for natural convection in heated cavity. Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

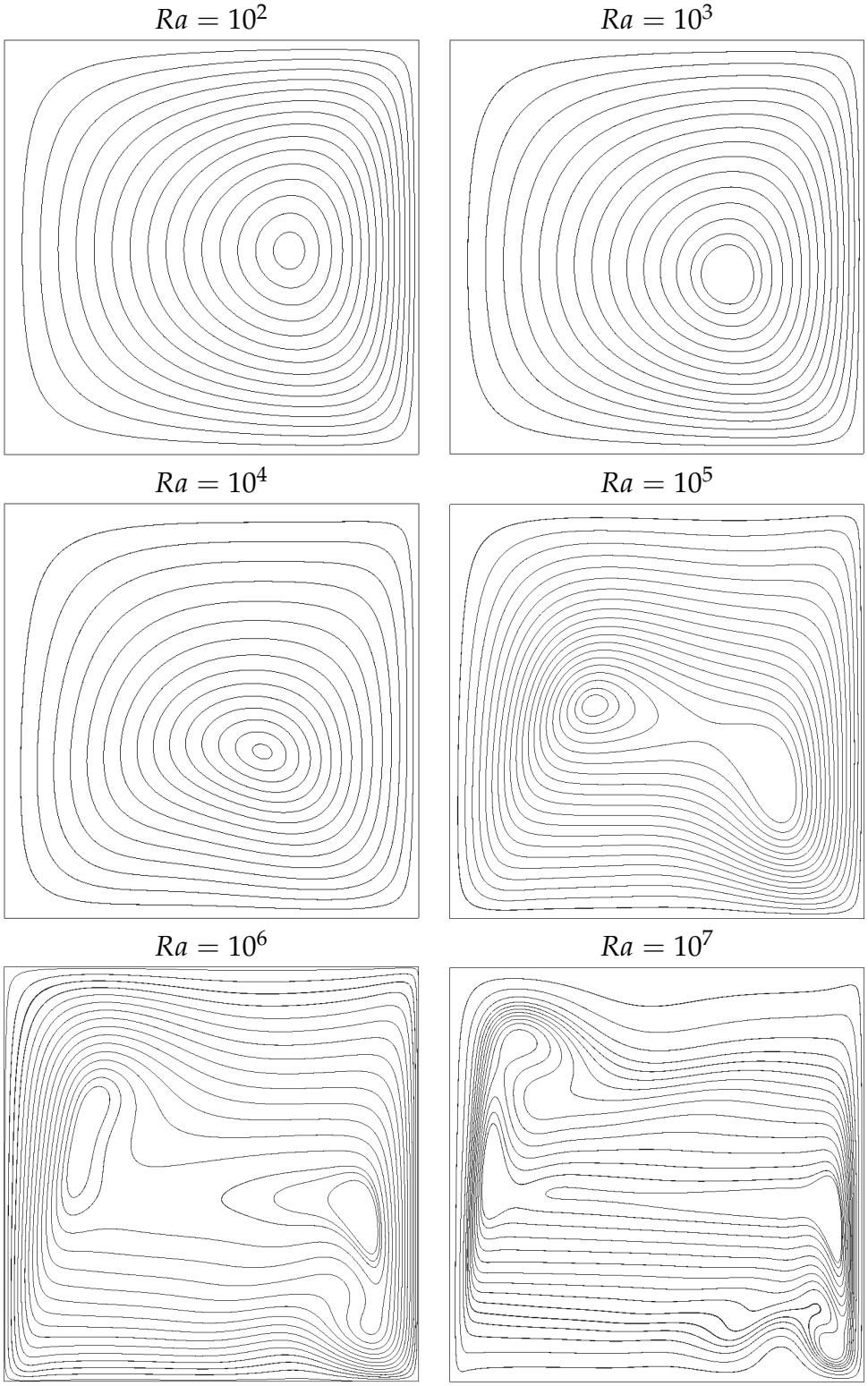


Figure 7.14: Streamlines for $Ra = 10^2 - 10^7$ for natural convection in heated cavity. Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

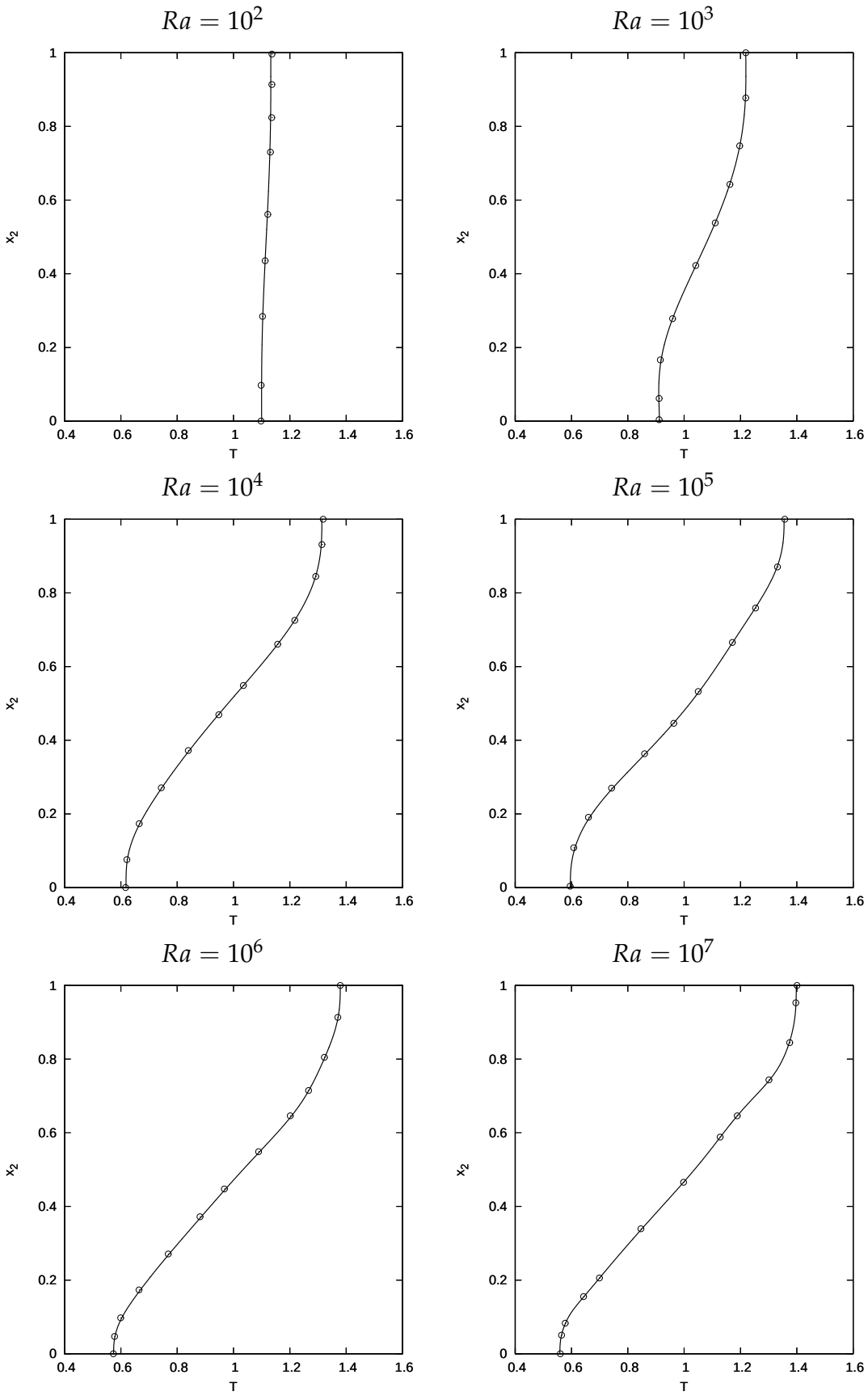


Figure 7.15: Temperature profiles for natural convection in heated cavity along vertical line $x_1 = 0.5$ - lines: present results, points: benchmark data from (Vierendeels et al. 2003). Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

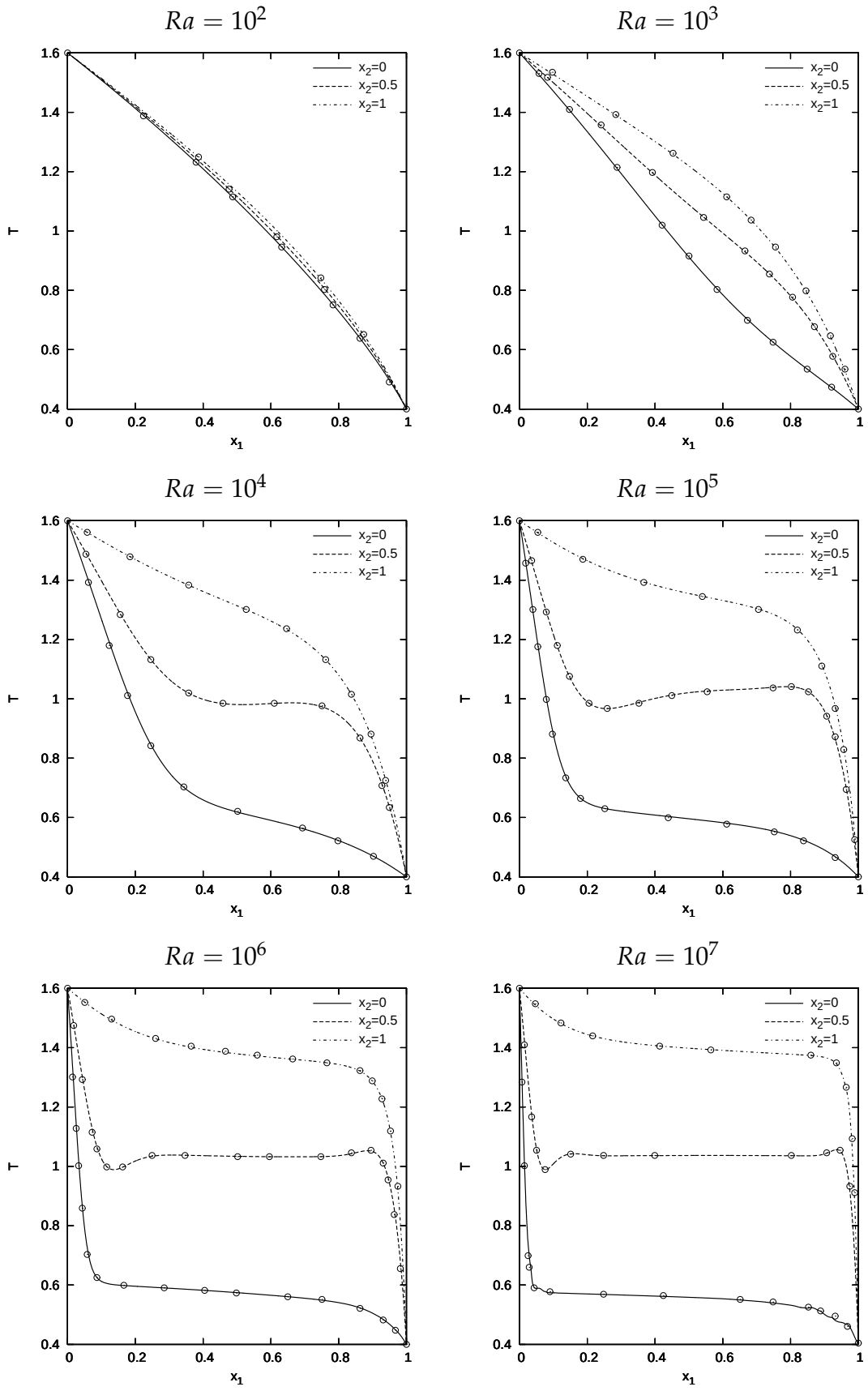


Figure 7.16: Temperature profiles for natural convection in heated cavity along three different horizontal lines - lines: present results, points: benchmark data from (Vierendeels et al. 2003). Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

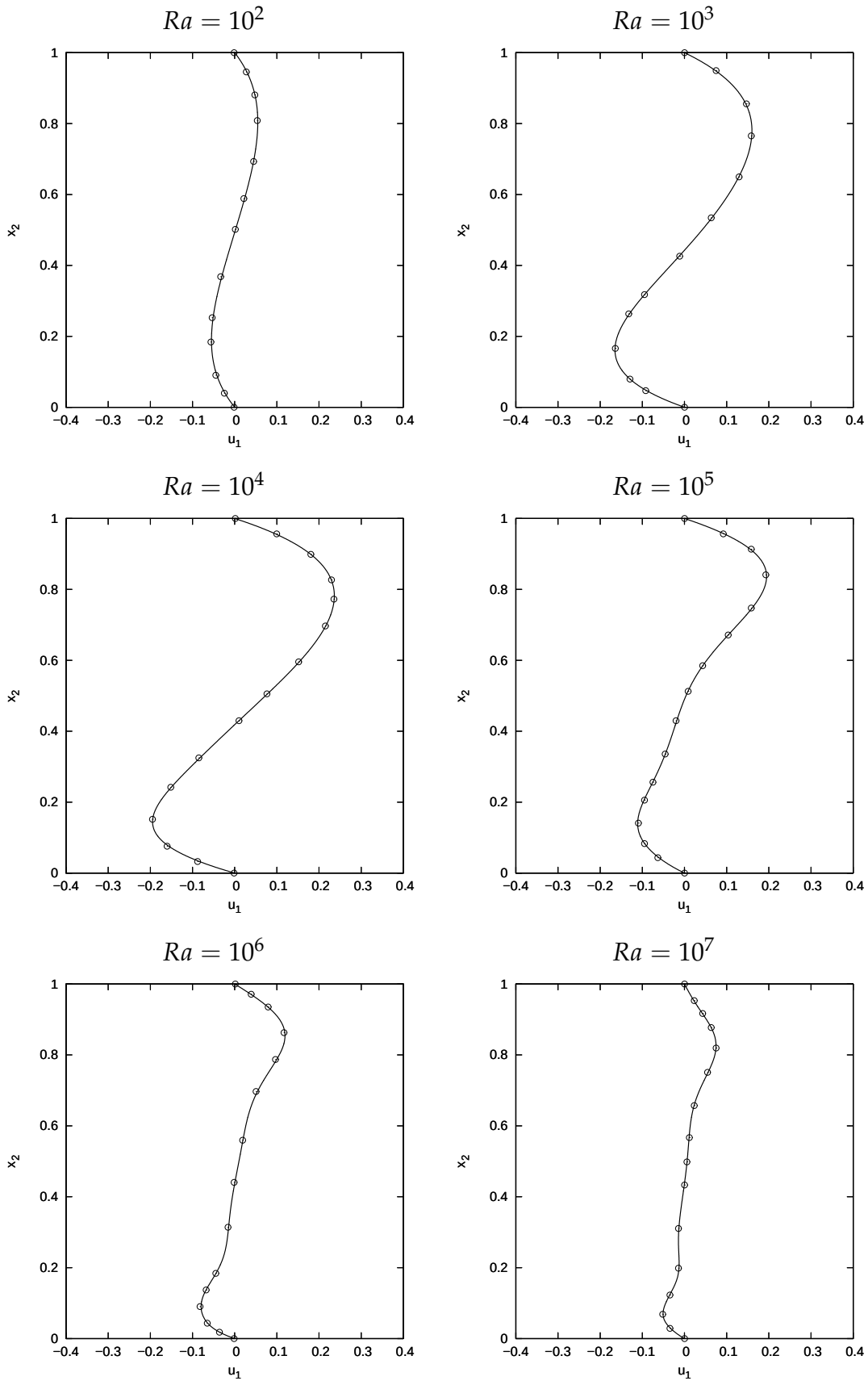


Figure 7.17: Velocity profile u_1 for natural convection in heated cavity along vertical line $x_1 = 0.5$ - lines: present results, points: benchmark data from (Vierendeels et al. 2003). Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

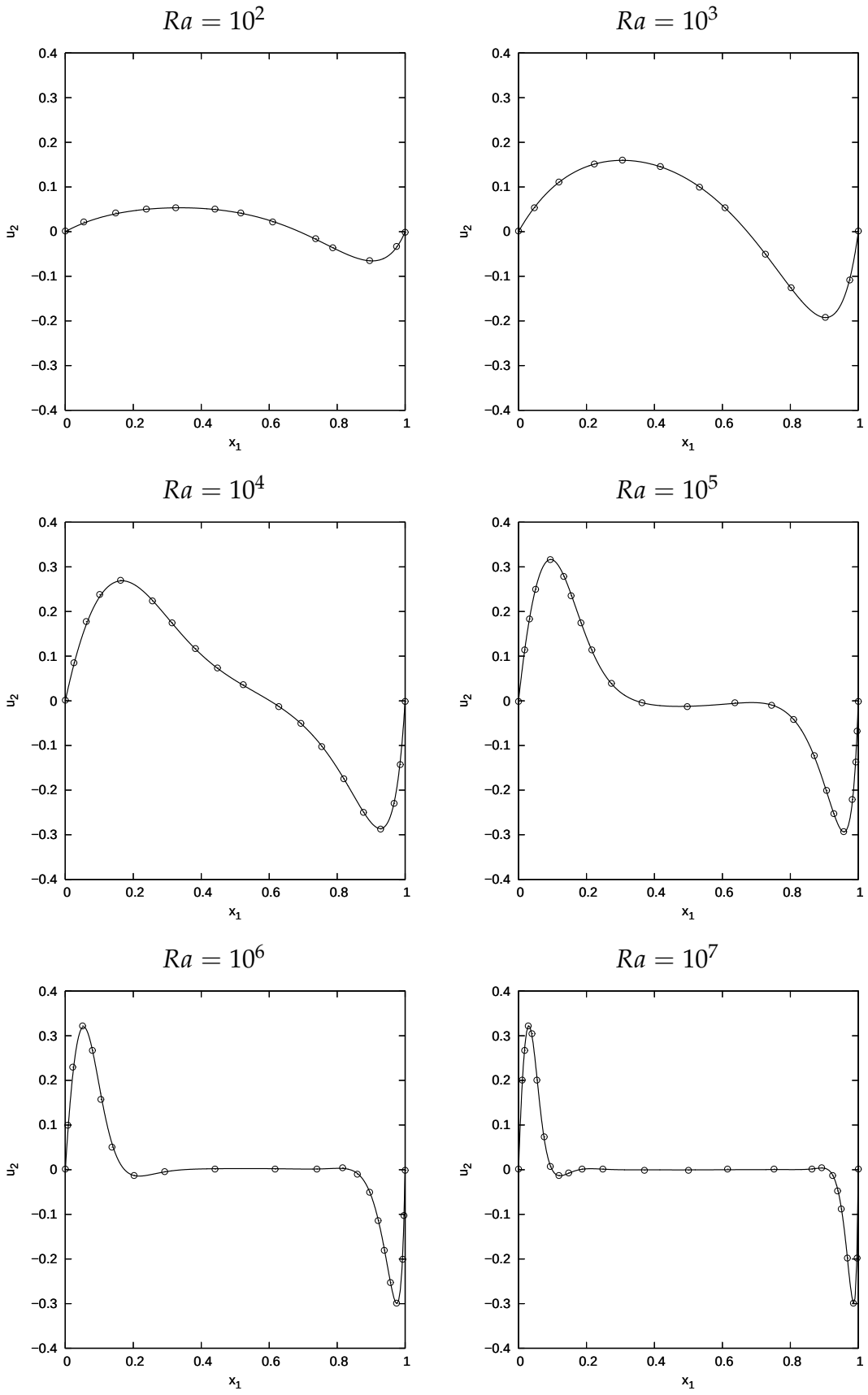


Figure 7.18: Velocity profile u_2 for natural convection in heated cavity along horizontal line $x_2 = 0.5$ - lines: present results, points: benchmark data from (Vierendeels et al. 2003). Results are calculated on a uniform Cartesian grid with 26×26 cells. Polynomial orders are 5 for velocity and temperature and 4 for pressure.

boundary condition (2.19b) for the temperature at the adiabatic walls, one obtains the following relation for the Nusselt numbers Nu_h and Nu_c at the hot and cold wall

$$\begin{aligned}\int_\Omega \frac{\partial \rho u_j T}{\partial x_j} - \frac{1}{RePr}\frac{\partial}{\partial x_j}\left(\lambda \frac{\partial T}{\partial x_j}\right) dx \\ &= \int_{\partial\Omega} \rho u_j T n_j - \frac{1}{RePr}\lambda \frac{\partial T}{\partial x_j} n_j ds \\ &= -\frac{1}{RePr}\int_{\partial\Omega_h} \lambda \frac{\partial T}{\partial x_1} dx_2 - \frac{1}{RePr}\int_{\partial\Omega_c} \lambda \frac{\partial T}{\partial x_1} dx_2 \\ &= -\frac{T_h - T_c}{RePr}(Nu_h + Nu_c) = 0. \end{aligned} \tag{7.29}$$

Note that even though the Discontinuous Galerkin formulation is conservative, the relation $Nu_h = -Nu_c$ (7.29) for the Nusselt numbers is not exactly fulfilled, because the boundary conditions are only imposed in the weak form. The error in the difference of both Nusselt numbers is a measure for the discretization error and tends to zero when increasing the spatial resolution.

In Table 7.2 the different polynomial orders and grids used for the accuracy assessment are summarized. The number of cells are chosen to yield approximately the same number of DOF for the different polynomial orders, i.e. we tackle the question whether refining the grid or increasing the polynomial order is more beneficial for this test case.

Table 7.2: Different polynomial orders, grids and according number of DOF (per variable) used for natural convection in heated cavity to assess the accuracy with respect to Nusselt number and thermodynamic pressure. The first number of polynomial orders is the degree of the velocity and the temperature and the second number of the pressure.

Polynomial order	Grid	DOF
2/1	50×50	15,000
3/2	38×38	14,440
4/3	31×31	14,415
5/4	26×26	14,196

In Fig. 7.19 the calculated Nusselt numbers at the hot and cold wall are plotted for the different combinations of polynomial orders and grids defined in Table 7.2 for all six Rayleigh numbers and compared to the reference values of (Vierendeels et al. 2003). For all Rayleigh numbers a difference in the two Nusselt numbers at the left and right wall can be observed, which is getting smaller when increasing the polynomial order and coarsening the grid at the same time. This difference between Nu_h and Nu_c is larger for higher Rayleigh numbers, which is reasonable since the flow structures are getting more complex. The agreement of the calculated Nusselt number at the hot wall Nu_h with the reference value is excellent. Especially for the higher Rayleigh numbers $Ra = 10^5 - 10^7$ the deviation of the Nusselt number Nu_c at the cold wall is larger compared with the benchmark solution, which is reasonable since the boundary layer

is thinner at the right wall and hence the gradients are steeper, see e.g. the profiles of the temperature and the velocity component u_2 along the horizontal line at $x_2 = 0.5$ in Fig. 7.16 and 7.18. In Fig. 7.20 the same study is done for the thermodynamic pressure. Again, the agreement with the reference values is much better for the simulations using polynomials of higher order on a coarser grid than vice versa. For $Ra = 10^2$ and $Ra = 10^3$ there seems to be a deviation from the reference value, which is not getting smaller for increasing the polynomial order. But this deviation is in the range of the fifth decimal digit, which is stated to be the accuracy of the reference solution. Also, the reference solution was computed based on the fully compressible Navier-Stokes equation, which might also have an impact on the last digits, cf. (Vierendeels et al. 2003).

The results for the Nusselt number and the thermodynamic pressure using the polynomial orders 5 for velocity and temperature and 4 for pressure on the grid with 26×26 cells are summarized in Table 7.3. For the higher Rayleigh numbers $Ra = 10^5 - 10^7$ our results are still slightly under resolved, but in general the implementation of the low-Mach number solver can be considered as verified. And like expected, for approximately the same number of DOF the simulations with higher order polynomials yield more accurate results than the ones with lower order polynomials.

Table 7.3: Comparison of Nusselt number and thermodynamic pressure with reference data from (Vierendeels et al. 2003) for natural convection in heated cavity using a grid with 26×26 cells and polynomial orders of 5 for velocity and temperature and 4 for pressure.

Rayleigh	Nu_h	Nu_c	Nu_{ref}	p_0	$p_{0,\text{ref}}$
10^2	0.9787	0.9787	0.9787	0.9574	0.95736
10^3	1.1077	1.1077	1.1077	0.9380	0.93805
10^4	2.2180	2.2171	2.2180	0.9146	0.91463
10^5	4.4801	4.4677	4.4800	0.9220	0.92196
10^6	8.6896	8.5800	8.6870	0.9245	0.92449
10^7	16.1799	15.6881	16.24	0.9222	0.92263

7.2.2.3 Performance

Finally, we show some exemplary results for the performance of the SIMPLE algorithm in Fig. 7.21. The pressure correction and the changes in the velocity components and the temperature are plotted versus the number of SIMPLE iterations for the different Rayleigh numbers using polynomial orders of 5 for velocity and temperature and 4 for pressure on the grid with 26×26 cells. The corresponding relaxation factors using the block diagonal option are those of Table 7.1. It is interesting to note that the number of SIMPLE iterations to reach the specified convergence criterion of 10^{-10} varies significantly for the different Rayleigh numbers, e.g. for $Ra = 10^2$ and $Ra = 10^3$ almost 8,000 iterations are needed, whereas for $Ra = 10^6$ the convergence criterion is reached already after about 1,700 iterations. This behavior is somewhat surprising

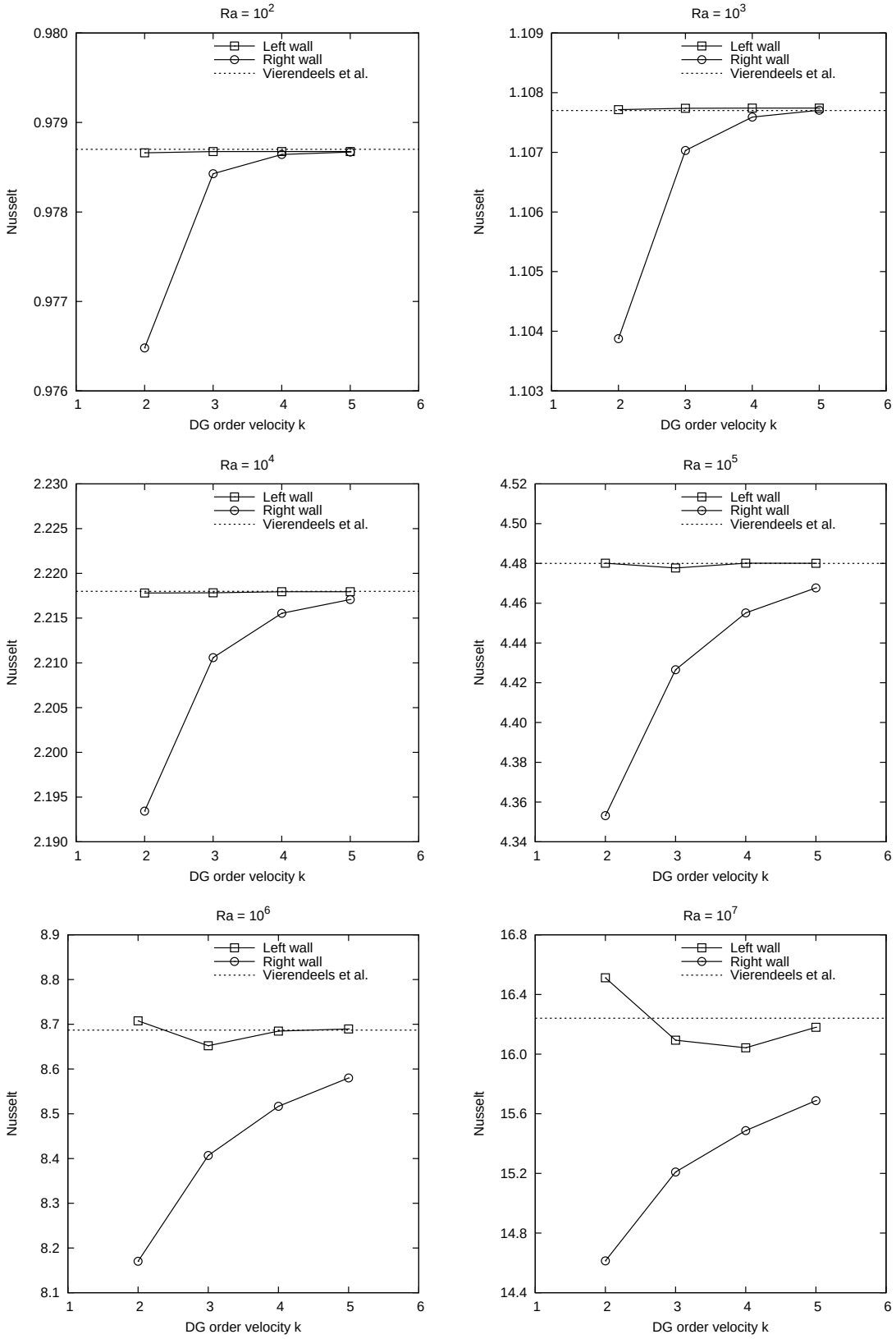


Figure 7.19: Calculated Nusselt numbers for natural convection in heated cavity at left (hot) and right (cold) wall for different Rayleigh numbers plotted versus polynomial order of velocity. For each polynomial order the number of grid cells is adjusted to yield approximately the same number of DOF, cf. Table 7.2. Reference values are taken from (Vierendeels et al. 2003).

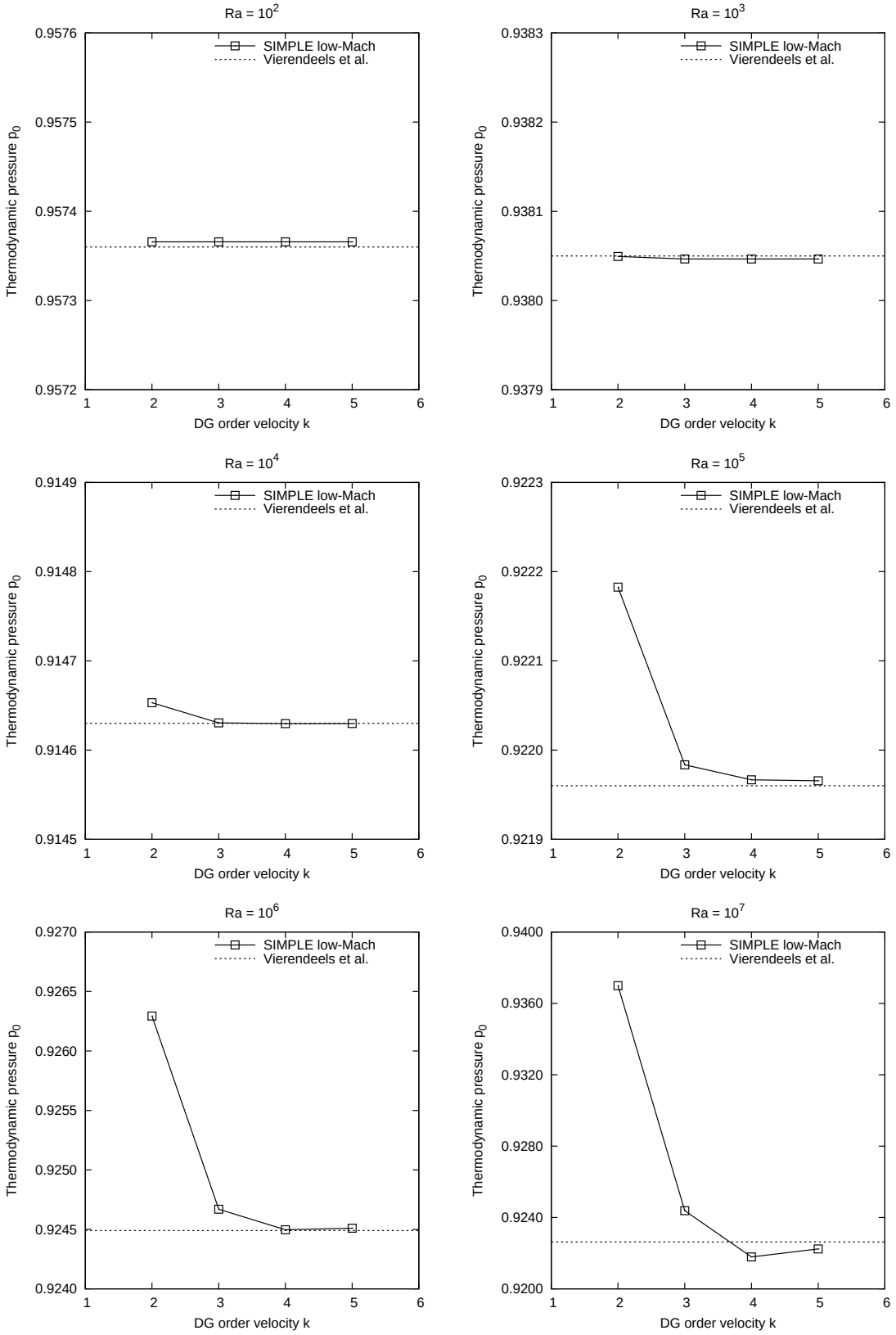


Figure 7.20: Calculated thermodynamic pressure for natural convection in heated cavity for different Rayleigh numbers plotted versus polynomial order of velocity. For each polynomial order the number of grid cells is adjusted to yield approximately the same number of DOF, cf. Table 7.2. Reference values are taken from (Vierendeels et al. 2003).

since the flow structures are more complex for the higher Rayleigh number case. We shall note that the relaxation factors for the simulations have not been optimized. A more detailed study using different combinations for the relaxation factors for velocity and pressure would be necessary to find the optimal relaxation factors for the different Rayleigh numbers. Overall, the convergence of the SIMPLE algorithm for this test case is quite satisfying. The convergence rate is almost constant down to nearly machine accuracy. Some of the reference solutions in (Le Quéré et al. 2005) have been computed using explicit or semi-implicit schemes in time, which were iterated until a steady state solution was obtained. It is stated that this approach can require up to hundreds of thousands of time steps, which demonstrates the benefits of our solution scheme solving the steady state equations for such cases.

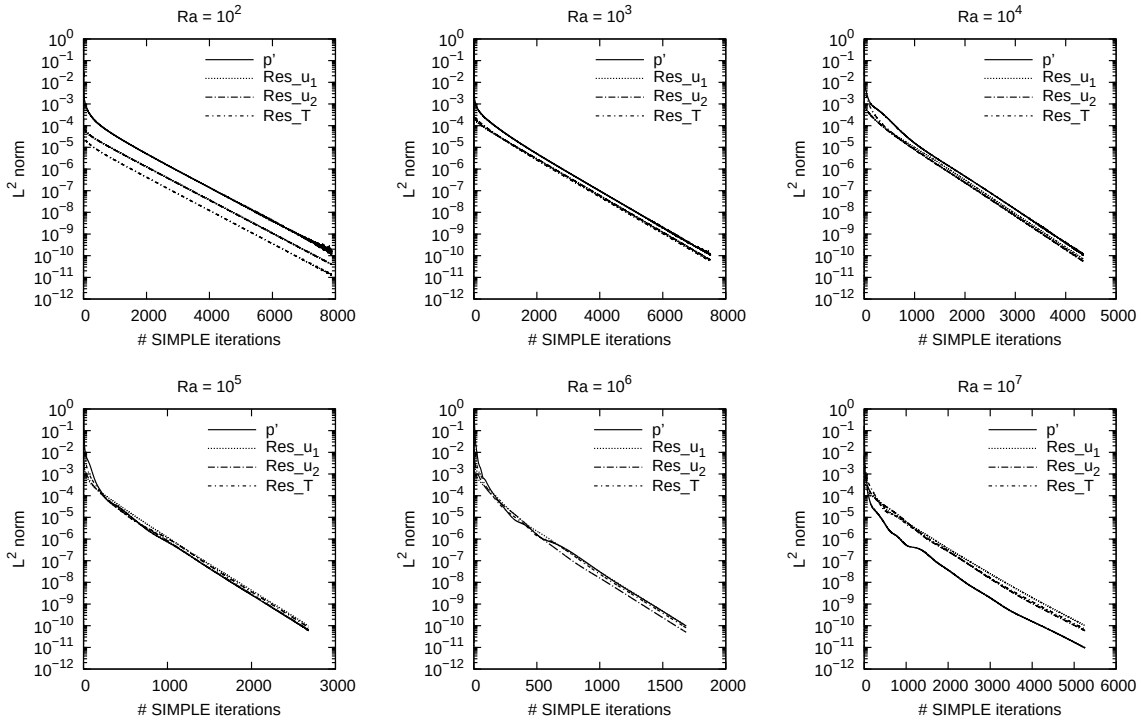


Figure 7.21: Performance of SIMPLE algorithm for natural convection in heated cavity for different Rayleigh numbers. Plotted are the pressure correction and the changes in the velocity components and the temperature versus the number of SIMPLE iterations using the block diagonal option. The grid consists of 26×26 cells and polynomial orders are 5 for velocity and temperature and 4 for pressure.

7.2.3 Unsteady natural convection in a tall cavity

This last test case for low-Mach number flows is similar to the previous one. The natural convection is simulated in a tall cavity with a height to width aspect ratio of 8:1. The geometry and the computational grid are shown in Fig. 7.22. Like for the previous test case, the fluid inside the cavity is subjected to a hot wall with temperature T_h and

a cold wall with temperature T_c at the left and right domain boundaries, respectively. The upper and lower domain boundaries are adiabatic walls and the gravity is directed downwards. The computational grid consists of 20×100 cells and a stretching is applied to refine the grid at the walls.

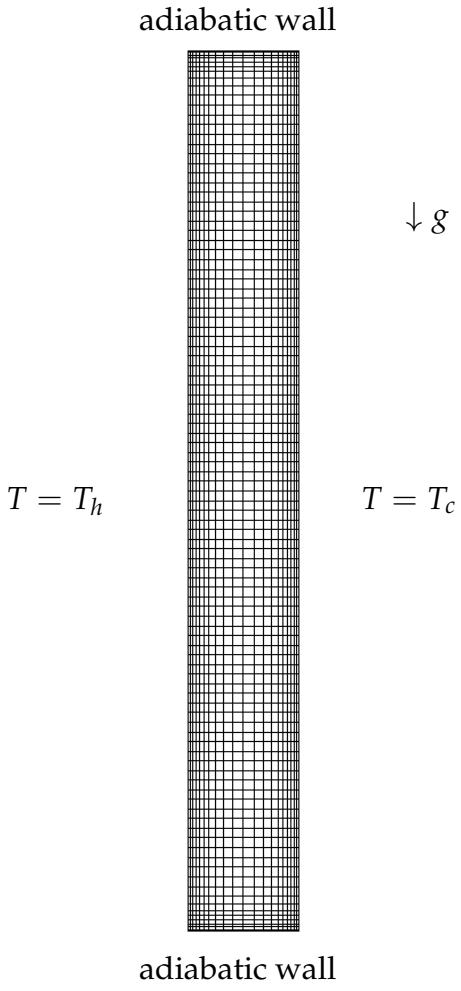


Figure 7.22: Geometry and computational grid with 20×100 cells for natural convection in a tall cavity with a height to width aspect ratio of 8:1. The top and bottom boundaries are adiabatic walls and at the left and right boundaries the temperature is prescribed. The no-slip condition applies to all domain boundaries.

Applying the Boussinesq approximation, extensive benchmark results for Prandtl number $Pr = 0.71$ and Rayleigh number $Ra = 3.4 \times 10^5$ are available in (Christon, Gresho & Sutton 2002). The chosen Rayleigh number is above the critical value, where unsteady periodic solutions are obtained. The setup for this test case is as follows:

- The Prandtl number is set to $Pr = 0.71$ and the heat capacity ratio to $\kappa = 1.4$.
- For Boussinesq flows the Rayleigh number is defined as

$$Ra = \frac{g\beta(\tilde{T}_h - \tilde{T}_c)\tilde{L}^3\tilde{\rho}_\infty^2\tilde{c_p}}{\tilde{\mu}_\infty\tilde{\lambda}_\infty}, \tag{7.30}$$

where L is the width of the cavity and β is the thermal expansion coefficient. For the reference velocity we take (cf. (Christon et al. 2002))

$$\tilde{u}_\infty = \left(g\beta(\tilde{T}_h - \tilde{T}_c)L\right)^{1/2}, \tag{7.31}$$

which yields the following relation for the Reynolds number

$$Re = \left(\frac{Ra}{Pr}\right)^{1/2}. \tag{7.32}$$

Taking $Pr = 0.71$ and $Ra = 3.4 \times 10^5$, we obtain a Reynolds number of $Re \approx 692.007$, which is used in the setup of our simulations.

- Since we solve the low-Mach number equations we need to choose a temperature difference

$$\epsilon = \frac{\tilde{T}_h - \tilde{T}_c}{2\tilde{T}_\infty}, \tag{7.33}$$

where $\tilde{T}_\infty = (\tilde{T}_h + \tilde{T}_c)/2$. Note that this is an additional parameter compared to the Boussinesq case, which is only valid for small temperature differences. To get comparable results to the benchmark solutions we set the temperature difference to $\epsilon = 0.1$, i.e. $T_h = 1.1$ and $T_c = 0.9$. Using the definitions above, the Froude number is given by

$$Fr = \left(\beta(\tilde{T}_h - \tilde{T}_c)\right)^{1/2}. \tag{7.34}$$

For ideal gases the thermal expansion coefficient is $\beta = 1/\tilde{T}_\infty$ leading to

$$Fr = (2\epsilon)^{1/2} \approx 0.447. \tag{7.35}$$

- The dynamic viscosity μ and the thermal conductivity λ are taken to be constant like in the Boussinesq case.

- Finally, the initial values are prescribed by

$$u_1(t = 0) = 0, \tag{7.36a}$$
$$u_2(t = 0) = 0, \tag{7.36b}$$
$$T(t = 0) = 1, \tag{7.36c}$$
$$p_0(t = 0) = 1. \tag{7.36d}$$

Next, we describe the solver setup for this test case. For the spatial discretization we apply polynomials of order 4 for velocity and temperature and 3 for pressure. Time discretization is done using the BDF-2 scheme with a time step size of $\Delta t = 0.075$. The block diagonal option is taken for the approximation in the corrector step. The relaxation factors are $\alpha_p = 0.2$ for pressure and $\alpha_u = 0.9$ for velocity. For the

temperature no under-relaxation is used. The SIMPLE iterations in each time step are stopped at a convergence criterion of 10^{-6}. For the pressure a reference point is set, since there is no pressure boundary condition.

In Fig. 7.23 we show the evolution of the temperature with time at the time-history point $(x_1, x_2) = (0.181, 7.37)$, which was defined in (Christon et al. 2002) for the benchmark solution. The temperature equivalent θ is defined as

$$\theta = \frac{\tilde{T} - \tilde{T}_\infty}{\tilde{T}_h - \tilde{T}_c}. \tag{7.37}$$

Starting from the initial conditions (7.36a)-(7.36d), we reach a time-periodic solution after a transitional phase, like expected. The periodic solution for a time interval of 10 periods is depicted in Fig. 7.24.

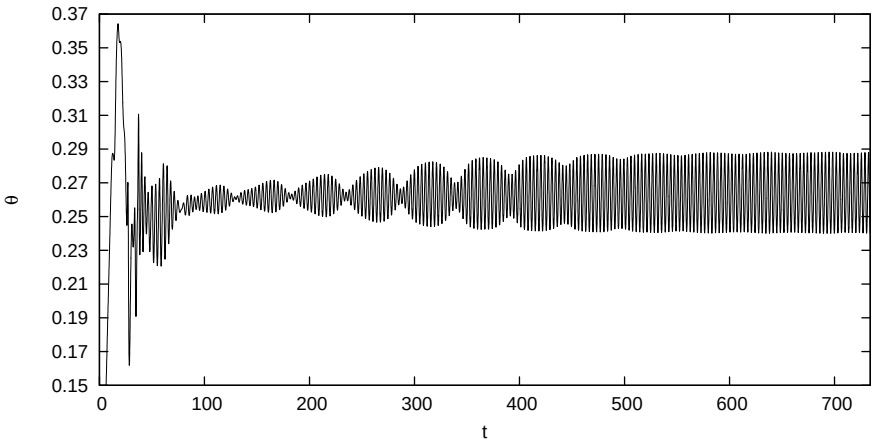


Figure 7.23: Temperature history for natural convection in tall cavity at point $(x_1, x_2) = (0.181, 7.37)$.

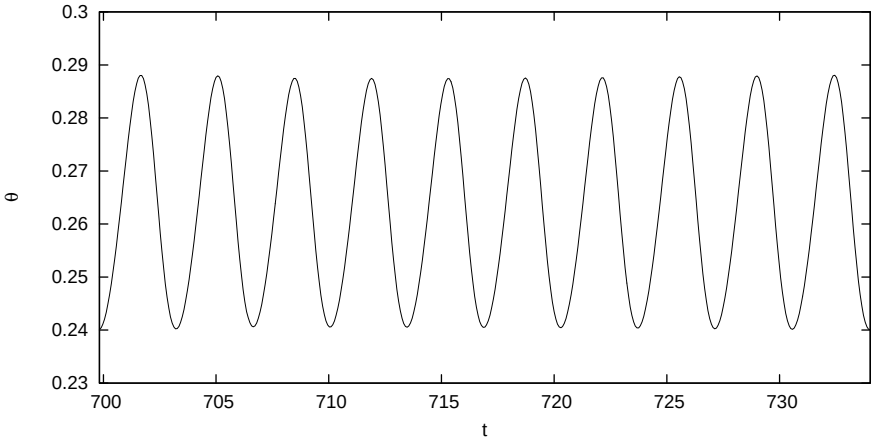


Figure 7.24: Temperature history for natural convection in tall cavity at point $(x_1, x_2) = (0.181, 7.37)$ for 10 periods after reaching a time-periodic state.

Instantaneous contours of the temperature, the velocity and the pressure after reaching the time-periodic state are plotted in Fig. 7.25. In the Boussinesq case the solution for the temperature and the velocity components are skew symmetric, cf. (Christon et al. 2002). Our results applying a temperature difference of $\epsilon = 0.1$ show already some deviation from this skew symmetry. This is consistent with the results reported in (Le Quéré, Masson & Perrot 1992) and (Bouloumou, Serre, Bontoux & Fröhlich 2012), where also the low-Mach number equations are solved for this test case. In both references it was observed that the symmetry is broken for higher values of the temperature difference. Still, our results are quite close to the benchmark solution for the Boussinesq case. In Table 7.4 we compare the mean value, the peak-to-valley oscillation amplitude and the frequency of the temperature equivalent θ in time-history point $(x_1, x_2) = (0.181, 7.37)$ as well as the mean values of the Nusselt number at the vertical walls with the reference values of (Xin & Le Quéré 2002). The results in (Xin & Le Quéré 2002) were simulated applying Chebyshev spatial approximations and were chosen as a baseline for comparison in (Christon et al. 2002) due to the high accuracy of the method. The mean values in Table 7.4 were averaged over 10 periods according to the time interval of Fig. 7.24. All values are found to be in good agreement with the reference values. Only the temperature amplitude is slightly higher in our case. Hence, the implementation of the unsteady low-Mach number solver can be considered to be verified.

Table 7.4: Comparison for natural convection in tall cavity with reference values. The quantities are the mean value, the peak-to-valley oscillation amplitude and the frequency of the temperature equivalent θ in time-history point $(x_1, x_2) = (0.181, 7.37)$ as well as the mean values of the Nusselt number at the vertical walls.

Reference	$\overline{\theta}_1$	θ_1'	τ_θ	$\overline{Nu}_h$	$\overline{Nu}_c$
Xin & Le Quéré (2002)	0.26548	0.04274	3.4115	4.57946	4.57946
Present	0.26289	0.04733	3.420	4.57858	4.57850

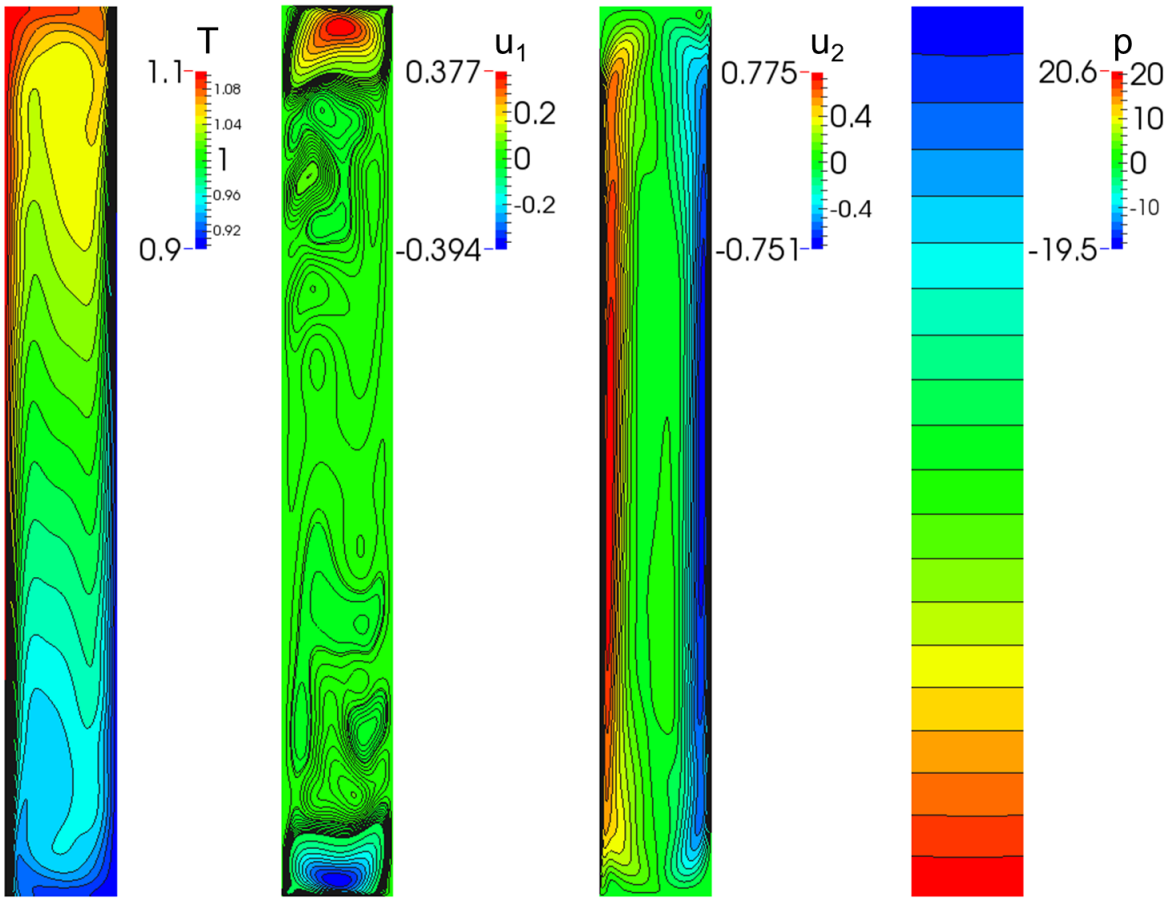


Figure 7.25: Instantaneous contour plots for natural convection in tall cavity after reaching the time-periodic state. Plotted are (from left to right): temperature, horizontal velocity component, vertical velocity component and pressure.

8 Conclusions and outlook

We presented a high-order DGM for the simulation of incompressible and variable density flows at low-Mach numbers. The SIMPLE algorithm was applied to solve the non-linear system of equations in an iterative and segregated manner for steady cases. The solver is implicit in time for unsteady cases using backward differentiation formulae, where also the SIMPLE algorithm was used to solve the non-linear systems in each time step. The developed method has been implemented in the in-house software library BoSSS, which is based on the DGM. The accuracy, performance and stability of the solver has been extensively investigated by simulating various test cases for incompressible and variable density flows at low-Mach numbers. The main results are summarized below.

We started by testing the solver for steady incompressible flows. Three different test cases have been considered: the Kovasznay flow, the flow into a corner and the backward-facing step flow. For the first two test cases analytical solutions of the incompressible Navier-Stokes equations are available, which were used for spatial convergence studies. Using a mixed-order formulation, i.e. order k for velocity and order $k-1$ for pressure, we obtained optimal convergence rates of $k+1$ for velocity and k for pressure. The results for the backward-facing step flow are in very good agreement with experimental results. The performance of the solver was tested for all three test cases with respect to the number of SIMPLE iterations to reach the specified convergence criterion. Applying the identity matrix for the approximation in the corrector step, the fastest convergence was achieved without under-relaxation for the pressure for all three test cases. Very recent tests, which have been performed in the context of a different project and therefore are not included in this thesis, have shown that the performance can be improved using diagonal or block diagonal matrices for the approximation in the corrector step, see also the results for variable density flows. In general, we observed that the performance of the solver is decreasing for finer meshes and polynomials of higher order, which is to be expected for a segregated solution algorithm. This suggests developing multigrid techniques to accelerate the performance of the algorithm.

The next step was the extension of the solver to unsteady incompressible flows. The unsteady test cases were the Taylor-vortex flow, the Orr-Sommerfeld stability problem and the flow past a square cylinder. We compared the mixed-order formulation, which was used before for the steady cases, with the equal-order formulation in terms of accuracy and stability. The equal-order formulation was stabilized by adding a penalty term for pressure jumps in the continuity equation, cf. (Cockburn et al. 2002, Cockburn et al. 2004, Cockburn et al. 2009). We also tested two approaches for the calculation of the pressure correction. The first one was the original formulation, which had been used before in the steady cases and we refer to as the SIMPLE-LDG method. The second

one was derived by replacing the original operator for the pressure correction with an equivalent operator stemming from the SIP method, hence we refer to this approach as the SIMPLE-SIP method. While the SIMPLE-LDG method involves the first and second layer of cell neighbors for the pressure correction, the SIMPLE-SIP method couples only the immediate neighbors. This idea stems from (Shahbazi et al. 2007), where a similar approach is used in the context of an algebraic splitting scheme. In a spatial convergence study for the Taylor-vortex flow, we obtained virtually the same absolute accuracy as well as convergence rates for velocity in pressure using the SIMPLE-LDG and the SIMPLE-SIP method. Applying the mixed-order formulation the convergence rates are $k+1$ and k for velocity and pressure, respectively. For the equal-order formulation the convergence rates are approximately the same, while the absolute accuracy is slightly better for the equal-order formulation. Comparing the performance of the different approaches in terms of SIMPLE iterations, we found that overall the mixed-order SIMPLE-LDG method performs best. The stability of the proposed method was assessed simulating the Orr-Sommerfeld stability problem for plane Poiseuille flow. We observed that the stability properties of the SIMPLE-LDG and SIMPLE-SIP method are practically the same. The stability of the mixed-order formulation is strongly dependent on the penalty parameter of the SIP method for the discretization of the viscous terms and the mesh size. We had to use an additional scaling factor for the penalty parameter to avoid unphysical instabilities. With this factor set to $\mu_{\mathrm{Sc}} = 5.0$ on a coarser grid and to $\mu_{\mathrm{Sc}} = 2.0$ on a finer grid, the numerical solution was in very good agreement with the results from linear stability analysis. For the error in the growth rate we obtained a spectral convergence rate with respect to the polynomial order k. The equal-order formulation was stable without adjusting the penalty parameter. Considering accuracy, performance, stability and mass conservation (i.e. a divergence-free velocity field), we conclude that the mixed-order SIMPLE-LDG method is the most valuable one. This method was also used to simulate the flow past a square cylinder. For two-dimensional flow at a Reynolds number of $Re = 100$ the results for Strouhal number and drag and lift coefficients are in very good agreement with experiments and numerical reference solutions. We also investigated the critical Reynolds numbers for the onset of vortex shedding as well as the transition from two- to three-dimensional flow, see also (Fischer 2014). The found values are in good agreement with the literature.

An intermediate step during the development of the solver for low-Mach number flows was a method for simulating multiphase flows with a smooth interface approach. Spatial discretization was done using a mixed-order formulation, i.e. order k for velocity and level set and $k-1$ for pressure. The first test case was the convection of a smooth density jump with a ratio of $\rho_1/\rho_2 = 1000$, which is quite a challenging test for numerical solvers. In a spatial convergence study we obtained convergence rates of approximately $k+1$ for the velocity and the level set function. The convergence rate for the pressure was above k, which is slightly higher than expected. The second test case was the two-phase Poiseuille flow with a horizontal interface at the center line of the channel. In a first step we applied a smooth interface approach for this test case. The solution perfectly matches the analytical solution. In a second step we used

a mesh, where the edges are aligned with the interface, to demonstrate the capabilities of the solver in a sharp interface framework. Therefore, we prescribed the level set function and only solved for the flow field variables, i.e. velocity and pressure. Using polynomial orders of 2 for velocity and 1 for pressure the numerical solution is exact up to machine accuracy and reproduces the kink in the velocity at the interface. We note that no surface tension and no reinitialization for the level set function was used in this work. This would be a starting point to extend the solver to more complex multiphase flow. Considering reinitialization it would be interesting to incorporate the method developed in (Olsson & Kreiss 2005) and (Olsson et al. 2007).

Finally, the developed solver was extended to simulate low-Mach number flows. To the best of the author's knowledge this is the first time that the DGM is applied to the low-Mach number equations. Again, we use a mixed-order formulation, i.e. order k for velocity and temperature and $k-1$ for pressure. The first test case was the Couette flow with a vertical temperature gradient. We considered constant material parameters for viscosity and thermal conductivity as well as variable parameters using the power law. In both cases we observe spatial convergence rates of approximately $k+1$ for velocity and temperature and k for pressure. The second test case was the natural convection in a heated cavity, which is a common benchmark problem for low-Mach number solvers. Simulations have been performed for six different Rayleigh numbers ranging from $Ra = 10^2$ to $Ra = 10^7$. Using a rather coarse grid with only 26×26 cells and polynomial orders of 5 for velocity and temperature and 4 for pressure the results for velocity and temperature profiles as well as for the Nusselt number and the thermodynamic pressure are in very good agreement with the reference values. Comparing the results with different combinations of grid resolution and polynomial order, where the number of DOF is approximately constant, we could demonstrate that in terms of accuracy it is beneficial to use coarse grids with polynomials of higher order. The last test case for low-Mach number flows was the natural convection in a tall cavity with a height to width aspect ratio of $8:1$. The Rayleigh number was set to $Ra = 3.4 \times 10^5$, for which an unsteady time-periodic solution is obtained. The results are in good agreement with benchmark data, which have been calculated using the Boussinesq approximation. Applying a temperature difference of $\epsilon = 0.1$ we observed deviations from the skew symmetry in the solution for the temperature and the velocity components, which has also been reported in the literature for solvers based on the low-Mach number equations rather than the Boussinesq approximation. Also for this test case we used a rather coarse grid with 20×100 cells and polynomial orders of 4 for velocity and temperature and 3 for pressure. The solver for low-Mach number flows was designed with the capability to be extended to combustion problems. It will be interesting to assess the capability of the DGM to handle reactive flows.

9 Bibliography

ARMALY, B. F., DURST, F., PEREIRA, J. C. F., SCHÖNUNG, B. (1983): Experimental and theoretical investigation of backward-facing step flow. *Journal of Fluid Mechanics 127*, 473–496.

ARNOLD, D. N., BREZZI, F., COCKBURN, B., MARINI, L. D. (2002): Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM Journal on Numerical Analysis 39*, 1749–1779.

BASSI, F., CRIVELLINI, A. (2006): A high-order discontinuous Galerkin method for natural convection problems. In: *Electronic Proceedings of the ECCOMAS CFD 2006 Conference*, P. Wesseling, E. Onate, J. Périaux (Eds.), Egmond aan Zee, The Netherlands, TU Delft.

BASSI, F., CRIVELLINI, A., DI PIETRO, D., REBAY, S. (2006): An artificial compressibility flux for the discontinuous Galerkin solution of the incompressible Navier–Stokes equations. *Journal of Computational Physics 218*, 2, 794–815.

BASSI, F., CRIVELLINI, A., DI PIETRO, D. A., REBAY, S. (2007): An implicit high-order discontinuous Galerkin method for steady and unsteady incompressible flows. *Computers & Fluids 36*, 1529–1546.

BASSI, F., DE BARTOLO, C., HARTMANN, R., NIGRO, A. (2009): A discontinuous Galerkin method for inviscid low Mach number flows. *Journal of Computational Physics 228*, 11, 3996–4011.

BECKER, R., BRAACK, M. (2002): Solution of a stationary benchmark problem for natural convection with large temperature difference. *International Journal of Thermal Sciences 41*, 5, 428–439.

BOULOUMOU, O., SERRE, E., BONTOUX, P., FRÖHLICH, J. (2012): A 3D pseudo-spectral low Mach-number solver for buoyancy driven flows with large temperature differences. *Computers & Fluids 66*, 107–120.

CHRISTON, M. A., GRESHO, P. M., SUTTON, S. B. (2002): Computational predictability of time-dependent natural convection flows in enclosures (including a benchmark solution). *International Journal for Numerical Methods in Fluids 40*, 8, 953–980.

COCKBURN, B. (2003): Discontinuous Galerkin methods. *ZAMM - Journal of Applied Mathematics and Mechanics / Zeitschrift für Angewandte Mathematik und Mechanik 83*, 11, 731–754.

COCKBURN, B., KANSCHAT, G., SCHÖTZAU, D. (2004): The Local Discontinuous Galerkin Method for the Oseen Equations. *Mathematics of Computation 73*, 246, 569–593.

COCKBURN, B., KANSCHAT, G., SCHÖTZAU, D. (2005): A locally conservative LDG method for the incompressible Navier-Stokes equations. *Mathematics of Computation 74*, 1067–1095.

COCKBURN, B., KANSCHAT, G., SCHÖTZAU, D. (2009): An equal-order DG method for the incompressible Navier-Stokes equations. *Journal of Scientific Computing 40*, 1-3, 188–210.

COCKBURN, B., KANSCHAT, G., SCHÖTZAU, D., SCHWAB, C. (2002): Local Discontinuous Galerkin Methods for the Stokes System. *SIAM Journal on Numerical Analysis 40*, 319.

COCKBURN, B., SHU, C. (1998): The local discontinuous Galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis 35*, 2440–2463.

COVELLO, V., NIGRO, A., DE BARTOLO, C., FLORIO, G. (2014): High-order discontinuous Galerkin solutions of internal low-Mach number turbulent flows. *Energy Procedia 45*, 528–537.

DAREKAR, R. M., SHERWIN, S. J. (2001): Flow past a square-section cylinder with a wavy stagnation face. *Journal of Fluid Mechanics 426*, 263–295.

DE VAHL DAVIS, G. (1983): Natural convection of air in a square cavity: A bench mark numerical solution. *International Journal for Numerical Methods in Fluids 3*, 3, 249–264.

DE VAHL DAVIS, G., JONES, I. P. (1983): Natural convection in a square cavity: A comparison exercise. *International Journal for Numerical Methods in Fluids 3*, 3, 227–248.

DRAZIN, P. G., RILEY, N. (2006): *The Navier-Stokes equations: a classification of flows and exact solutions*. No. 334 in London Mathematical Society Lecture Note Series, Cambridge University Press.

EMAMY, N. (2014): *Numerical simulation of deformation of a droplet in a stationary electric field using DG*. Ph.D. thesis, TU Darmstadt.

FALGOUT, R., JONES, J., YANG, U. (2006): The design and implementation of hypre, a library of parallel high performance preconditioners. In: *Numerical Solution of Partial Differential Equations on Parallel Computers*, A. Bruaset, A. Tveito, eds., Springer, vol. 51, 267–294.

FEISTAUER, M., KUČERA, V. (2007): On a robust discontinuous Galerkin technique for the solution of compressible flow. *Journal of Computational Physics 224*, 1, 208–221.

FERRER, E., WILLDEN, R. H. J. (2011): A high order Discontinuous Galerkin Finite Element solver for the incompressible Navier-Stokes equations. *Computers & Fluids 46*, 224–230.

FERZIGER, J. H., PERIĆ, M. (2002): *Computational methods for fluid dynamics*. Springer, Third edn..

FISCHER, M. (2014): *Verification and validation of a discontinuous Galerkin based CFD solver*. Master thesis, Technische Universität Darmstadt.

GIRAULT, V., RIVIÈRE, B., WHEELER, M. F. (2005): A splitting method using discontinuous Galerkin for the transient incompressible Navier-Stokes equations. *ESAIM: Mathematical Modelling and Numerical Analysis 39*, 1115–1147.

GRESHO, P. M. (1991): Incompressible fluid dynamics: Some fundamental formulation issues. *Annual Review of Fluid Mechanics 23*, 413–453.

GROOSS, J., HESTHAVEN, J. (2006): A level set discontinuous Galerkin method for free surface flows. *Computer Methods in Applied Mechanics and Engineering 195*, 25–28, 3406–3429.

HAHN, F. (2009): *Zur Vorhersage technischer Verbrennungssysteme im Hinblickauf flüssige Brennstoffe*. Ph.D. thesis, Technische Universität Darmstadt, VDI Verlag GmbH, Düsseldorf.

HAIRER, E., WANNER, G. (1996): *Solving Ordinary Differential Equations II*. Springer, Second edn..

HAROUTUNIAN, V., ENGELMAN, M. S., HASBANI, I. (1993): Segregated finite element algorithms for the numerical solution of large-scale incompressible flow problems. *International Journal for Numerical Methods in Fluids 17*, 323–348.

HEIMANN, F., ENGWER, C., IPPISCH, O., BASTIAN, P. (2013): An unfitted interior penalty discontinuous Galerkin method for incompressible Navier–Stokes two-phase flow. *International Journal for Numerical Methods in Fluids 71*, 3, 269–293.

HESTHAVEN, J. S., WARBURTON, T. (2007): *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer US, First edn..

HILLEWAERT, K. (2013): *Development of the discontinuous Galerkin method for high-resolution, large scale CFD and acoustics in industrial geometries*. Ph.D. thesis, Université catholique de Louvain.

JOHNSON, C., PITKÄRANTA, J. (1986): An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Mathematics of Computation 46*, 173, 1–26.

KEIL, M. (2012): *Erweiterung und Validierung eines stationären SIMPLE-Lösers für zeitabhängige Probleme*. Bachelor thesis, Technische Universität Darmstadt.

KLEIN, B. (2011): *Implementation of the SIMPLE algorithm for solving the steady incompressible Navier-Stokes equations discretized by the discontinuous Galerkin method*. Diploma thesis, Technische Universität Darmstadt.

KLEIN, B., KUMMER, F., KEIL, M., OBERLACK, M. (2015): An extension of the SIMPLE based discontinuous Galerkin solver to unsteady incompressible flows. *International Journal for Numerical Methods in Fluids 77*, 10, 571–589.

KLEIN, B., KUMMER, F., OBERLACK, M. (2013): A SIMPLE based discontinuous Galerkin solver for steady incompressible flows. *Journal of Computational Physics 237*, 235–250.

KOVASZNAY, L. I. G. (1948): Laminar flow behind a two-dimensional grid. *Mathematical Proceedings of the Cambridge Philosophical Society 44*, 58–62.

KUMMER, F. (2012): *The BoSSS Discontinuous Galerkin solver for incompressible fluid dynamics and an extension to singular equations*. Ph.D. thesis, Technische Universität Darmstadt.

KUMMER, F. (2013): Extended Discontiuous Galerkin methods for multiphase flows: the spatial discretization. Annual research briefs 2013, Center for Turbulence Research.

KUMMER, F., EMAMY, N., MOUSAVI BELFEH TEYMOURI, R., OBERLACK, M. (2009): Report on the development of a generic discontinuous Galerkin framework in .NET. In: *ParCFD 2009, 21st International Conference on Parallel Computational Fluid Dynamics, May 18-22, 2009, Moffett Field, California, USA*.

KUMMER, F., OBERLACK, M. (2013): An extension of the Discontinuous Galerkin Method for the singular Poisson equation. *SIAM Journal on Scientific Computing 35*, 2, A603–A622.

LE QUÉRÉ, P., MASSON, R., PERROT, P. (1992): A Chebyshev collocation algorithm for 2D non-Boussinesq convection. *Journal of Computational Physics 103*, 2, 320–335.

LE QUÉRÉ, P., WEISMAN, C., PAILLÈRE, H., VIERENDEELS, J., DICK, E., BECKER, R., BRAACK, M., LOCKE, J. (2005): Modelling of natural convection flows with large temperature differences: A benchmark problem for low Mach number solvers. Part 1. Reference solutions. *ESAIM: Mathematical Modelling and Numerical Analysis 39*, 3, 609–616.

LI, B. Q. (2006): *Discontinuous finite elements in fluid dynamics and heat transfer*. Springer, London.

LUO, H., BAUM, J. D., LÖHNER, R. (2006): A fast, p-Multigrid discontinuous Galerkin method for compressible flows at all speeds. In: *American Institute of Aeronautics and Astronautics Paper*, Reno, Nevada.

MAJDA, A., SETHIAN, J. (1985): The derivation and numerical solution of the equations for zero Mach number combustion. *Combustion Science and Technology 42*, 3-4, 185–205.

MALIK, M. R., ZANG, T. A., HUSSAINI, M. Y. (1985): A spectral collocation method for the Navier-Stokes equations. *Journal of Computational Physics 61*, 1, 64–88.

MÜLLER, B. (2014): *Methods for higher order numerical simulations of complex inviscid fluids with immersed boundaries*. Ph.D. thesis, Technische Universität Darmstadt.

MOUSAVI, R. (2014): *Level set method for simulating the dynamics of the fluid-fluid interfaces: Application of a Discontinuous Galerkin Method*. Ph.D. thesis, Technische Universität Darmstadt.

NIGRO, A., DE BARTOLO, C., HARTMANN, R., BASSI, F. (2010): Discontinuous Galerkin solution of preconditioned Euler equations for very low Mach number flows. *International Journal for Numerical Methods in Fluids 63*, 4, 449–467.

NIGRO, A., RENDA, S., DE BARTOLO, C., HARTMANN, R., BASSI, F. (2013): A high-order accurate discontinuous Galerkin finite element method for laminar low Mach number flows. *International Journal for Numerical Methods in Fluids 72*, 1, 43–68.

OKAJIMA, A. (1982): Strouhal numbers of rectangular cylinders. *Journal of Fluid Mechanics 123*, 379–398.

OLBRICHT, C. (2009): *Numerische Berechnung technischer Verbrennungssysteme*. Ph.D. thesis, Technische Universität Darmstadt, VDI Verlag GmbH, Düsseldorf.

OLSSON, E., KREISS, G. (2005): A conservative level set method for two phase flow. *Journal of Computational Physics 210*, 1, 225–246.

OLSSON, E., KREISS, G., ZAHEDI, S. (2007): A conservative level set method for two phase flow II. *Journal of Computational Physics 225*, 1, 785–807.

OWKES, M., DESJARDINS, O. (2013): A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. *Journal of Computational Physics 249*, 275–302.

PAILLÈRE, H., LE QUÉRÉ, P., WEISMAN, C., VIERENDEELS, J., DICK, E., BRAACK, M., DABBENE, F., BECCANTINI, A., STUDER, E., KLOCZKO, T., CORRE, C., HEUVELINE, V., DARBANDI, M., HOSSEINIZADEH, S. (2005): Modelling of natural convection flows with large temperature differences: A benchmark problem for low Mach number solvers. Part 2. Contributions to the June 2004 conference. *ESAIM: Mathematical Modelling and Numerical Analysis 39*, 03, 617–621.

PATANKAR, S. V., SPALDING, D. B. (1972): A calculation procedure for heat, mass and momentum transfer in three-dimensional parabolic flows. *International Journal of Heat and Mass Transfer 15*, 1787–1806.

PERSSON, P.-O., PERAIRE, J. (2008): Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. *SIAM Journal on Scientific Computing 30*, 6, 2709–2733.

POCHET, F., HILLEWAERT, K., GEUZAINE, P., REMACLE, J.-F., MARCHANDISE, . (2013): A 3D strongly coupled implicit discontinuous Galerkin level set-based method for modeling two-phase flows. *Computers & Fluids* , 144–155.

RAUWOENS, P., VIERENDEELS, J., DICK, E., MERCI, B. (2009): A conservative discrete compatibility-constraint low-Mach pressure-correction algorithm for time-

accurate simulations of variable density flows. *Journal of Computational Physics 228*, 13, 4714–4744.

RENDA, S. M., HARTMANN, R., DE BARTOLO, C., WALLRAFF, M. (2015): A high-order discontinuous Galerkin method for all-speed flows. *International Journal for Numerical Methods in Fluids 77*, 4, 224–247.

RHEBERGEN, S., COCKBURN, B., VAN DER VEGT, J. (2013): A space-time discontinuous Galerkin method for the incompressible Navier-Stokes equations. *Journal of Computational Physics 233*, 339–358.

RIVIÈRE, B., GIRAULT, V. (2006): Discontinuous finite element methods for incompressible flows on subdomains with non-matching interfaces. *Computer Methods in Applied Mechanics and Engineering 195*, 3274–3292.

ROOK, R. (2001): *Acoustics in burner-stabilised flames*. Ph.D. thesis, Technische Universiteit Eindhoven.

SCHENK, O., GÄRTNER, K. (2004): Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Generation Computer Systems 20*, 3, 475–487.

SCHENK, O., GÄRTNER, K. (2006): On fast factorization pivoting methods for sparse symmetric indefinite systems. *ETNA. Electronic Transactions on Numerical Analysis [electronic only] 23*, 158–179.

SHAHBAZI, K. (2005): An explicit expression for the penalty parameter of the interior penalty method. *Journal of Computational Physics 205*, 401–407.

SHAHBAZI, K., FISCHER, P. F., ETHIER, C. R. (2007): A high-order discontinuous Galerkin method for the unsteady incompressible Navier-Stokes equations. *Journal of Computational Physics 222*, 391–407.

TAVELLI, M., DUMBSER, M. (2014): A staggered semi-implicit discontinuous Galerkin method for the two dimensional incompressible Navier–Stokes equations. *Applied Mathematics and Computation 248*, 70–92.

UTZ, T. (2014): *Validation of curved elements for incompressible, viscous flow problems*. Master thesis, Technische Universität Darmstadt.

VIERENDEELS, J., MERCI, B., DICK, E. (2003): Benchmark solutions for the natural convective heat transfer problem in a square cavity with large horizontal temperature differences. *International Journal of Numerical Methods for Heat & Fluid Flow 13*, 8, 1057–1078.

WANG, Y., OBERLACK, M. (2011): A thermodynamic model of multiphase flows with moving interfaces and contact line. *Continuum Mechanics and Thermodynamics 23*, 5, 409–433.

WANG, Z., FIDKOWSKI, K., ABGRALL, R., BASSI, F., CARAENI, D., CARY, A., DECONINCK, H., HARTMANN, R., HILLEWAERT, K., HUYNH, H., KROLL, N., MAY, G., PERSSON, P.-O., VAN LEER, B., VISBAL, M. (2013): High-order CFD methods:

current status and perspective. *International Journal for Numerical Methods in Fluids 72*, 8, 811–845.

WEI, L., POLLARD, A. (2011): Direct numerical simulation of compressible turbulent channel flows using the discontinuous Galerkin method. *Computers & Fluids 47*, 1, 85–100.

XIN, S., LE QUÉRÉ, P. (2002): An extended Chebyshev pseudo-spectral benchmark for the 8:1 differentially heated cavity. *International Journal for Numerical Methods in Fluids 40*, 8, 981–998.